Trust in Decision Aids: A Model and a Training Strategy

Book · January 1997

CITATIONS
13

READS
914

4 authors, including:

Marvin Cohen
Perceptronics Solutions
100 PUBLICATIONS **1,220** CITATIONS
SEE PROFILE

Raja Parasuraman
George Mason University
317 PUBLICATIONS **31,682** CITATIONS
SEE PROFILE

Daniel Serfaty
Aptima-Human Centered Engineering
86 PUBLICATIONS **1,596** CITATIONS
SEE PROFILE

USAATCOM TR 97-D-4

**U.S. ARMY AVIATION
AND TROOP COMMAND**

Trust in Decision Aids: A Model and a Training Strategy

**MARVIN S. COHEN, RAJA PARASURAMAN, DANIEL SERFATY, AND
ROBERT C. ANDES**

**COGNITIVE TECHNOLOGIES, INC.
4200 LORCOM LANE
ARLINGTON, VA 22207**

NOVEMBER 1997

FINAL REPORT

Prepared for

**AVIATION APPLIED TECHNOLOGY DIRECTORATE
AVIATION RESEARCH, DEVELOPMENT & ENGINEERING CENTER (ATCOM)
FORT EUSTIS, VA 23604-5577**

USAATCOM TR 97-D-4

U.S. ARMY AVIATION
AND TROOP COMMAND

Trust in Decision Aids: A Model and a Training Strategy

MARVIN S. COHEN, RAJA PARASURAMAN, DANIEL SERFATY, AND
ROBERT C. ANDES

NOVEMBER 1997

FINAL REPORT

SBIR DATA RIGHTS

Contract No. DAAJ02-97-C-0009
Contractor Name: Cognitive Technologies, Inc.
Address: 4200 Lorcom Lane, Arlington, VA 22207
Expiration of SBIR Data Rights Period: June 2002

Prepared for

AVIATION APPLIED TECHNOLOGY DIRECTORATE
AVIATION RESEARCH, DEVELOPMENT & ENGINEERING CENTER (ATCOM)
FORT EUSTIS, VA 23604-5577

ACKNOWLEDGMENTS

We are grateful to project's technical monitor, Keith Arthur, for his invaluable guidance and support throughout this project. Thanks also to the scientists at McDonnell Douglas Helicopter Systems, especially Greta Robertson, Bill Baker, and Ken Wroblewski, who generously allowed us to attend Rotorcraft Pilot's Associate briefings and tests. Last but not least, we thank Ed Lee, Kent A. Knapp, John Vandenberg, and Richard Perszyk for their valuable feedback and comments on the training package.

CONTENTS

TABLES

FIGURES

INTRODUCTION

The Problem of Automated Decision Making

There is considerable interest in the development of computerized aids to support, assist, or in part even replace human decision makers. Such efforts will increase in number with the growing digitization of the battlefield. Decision aids have much in common with other types of automation. For example, they vary in the *level of automation* that they offer (Parasuraman & Riley, 1997; Sheridan, 1992) — from data integration, through expert systems that offer decision options, to associate systems that take action unless overridden by the user. In principle, "autonomous" systems can be developed that choose and act without informing the user.

Unlike automation in other fields, however, automation of decision making has not advanced as far along this spectrum as some have hoped. An important reason for this is a lack of *trust* in computer decisions for most real-world problems. In particular, for example, trust is central to the effectiveness of middle-level decision aids such as expert and associate systems (Lee & Moray, 1992; Roth, Bennett, & Woods, 1988). A decision aid can be less helpful than the designer predicted either because human users do not employ the decision aid when they should, or because users rely on the aid when they should not. Behind these problems of under-trust and over-trust, respectively, is a failure to understand or properly evaluate the basis for the aid's decisions, i.e., an incomplete mental model of the aid (Parasuraman & Riley, 1997).

This discrepancy between designers' intent and real-world results highlights a difference between decision aids and other types of automation. Human decision making under uncertainty is imperfect, and decision aids are intended to help users handle uncertainty. Yet, an obstacle to the effective use of decision aids is uncertainty about the decision aids themselves. Unlike other kinds of automation, therefore, decision aids do not fully free users from the task that was to have been automated. The distinctiveness of decision aiding as a form of automation has important implications for training. To benefit from a decision aid, the user must learn to assess and act on uncertainty about the quality of the aid's recommendations. In most cases, this task is not likely to be trivial, or to be resolved over a small number of experiences with the aid. The domains in which decision aids are introduced tend to be complex; novel situations are likely to arise that were not anticipated by aid designers; and because of uncertainty, even the best course of action may on occasion have a bad outcome, or a bad course of action a good outcome. It is not easy in such a domain to acquire an understanding of the aid's decision making processes that will support effective exploitation of the aid.

Slowness in acceptance of decision aids can be traced in part to mismatches of aid design with human cognitive preferences and aptitudes, with organizational culture, or with management style. A significant share of the blame, however, must be placed on training, and in particular, on the failure of training to address the issue of trust. Existing training focuses on making the aid work, i.e., on how to input required information, how to change modes of aid operation, and how to use aid outputs. There is very little effort to improve skills for evaluating an aid's performance or to teach strategies for using the aid based on that evaluation.

A growing body of empirical and analytical research has examined the factors associated with effective and ineffective use of automation by human decision makers (e.g., for a review, see Parasuraman & Mouloua, 1996). In parallel, there is an increasing understanding of how proficient decision makers actually make decisions in real-world domains, and how less experienced decision makers can be effectively trained to make better decisions (Cohen, Freeman, & Thompson, 1997). These lines of inquiry need to be combined, to gain a better understanding of the distinctive decision processes underlying successful interaction with a decision aid. The result can serve as a foundation for developing training interventions that promote more effective decisions about the use of decision aids.

Overview

This research had three goals:

- To develop a systematic and general framework for training users of decision aids.
- To apply the framework and test its feasibility by developing a training strategy for a specific decision aiding environment — i.e., the Rotorcraft Pilot's Associate.
- To identify methods for measuring effects of decision aiding in terms of workload and situation awareness.

The present report describes the results of all three goals at the conclusion of a six-month Phase I Small Business Innovative Research project.

We address the first objective, the development of a systematic training framework, in three stages in the next three chapters. Each stage adds a piece of the framework, and draws implications for training users of decision aids from

that piece. At the same time, each stage becomes an ingredient for the development of a larger piece of the framework in the following chapter.

The first step (Chapter 0) is the development of a model of a user's trust in a decision aid that accounts for the findings of recent empirical research in a rigorous, consistent, and pragmatically useful way. The model has both a qualitative aspect, based on the structure of arguments about expected system performance, and a quantitative aspect, based on the probability calculus. With respect to training, this model generates a variety of concepts for what training of decision aid users should attempt to convey: It provides an account of the mental models required by savvy decision aid users at each phase of decision aid use, and the monitoring and situation awareness skills required to exploit those mental models effectively. It gives a description of the assessment skills that build upon those mental models to predict system performance. And it provides insight into critical thinking skills necessary to identify and handle novel situations. In addition, the trust model provides tools for generating scenarios in which users can acquire the relevant mental models, and diagnostic measures to assess their progress and the effectiveness of the training.

The second step in the construction of the framework (Chapter 0 and Appendix A) is the utilization of the trust model as an input for a set of models of user-decision aid interaction. These models clarify the strategies available to decision aid users and others at different phases of decision aid use, for example, decisions about system automation capabilities at the design stage, user selection of automation mode during mission planning, and user compliance or non-compliance with an aid's recommendations during mission execution. With respect to training, these benchmark models help define appropriate strategies for decision aid use, as a function of trust in the aid, the time available to the user, and the importance of the task. They provide the tools for creating rich scenarios in which such strategies can be elicited and practiced, and they provide the basis for clear and appropriate feedback.

The third step in the development of the training framework (Chapter 0 and Appendix B) is a classification of user-decision aid interactions, and pitfalls associated with them. The classification is derived by crossing decision aid functions (such as situation assessment and option generation) with user supervisory tasks (such as selecting automation mode and monitoring the aid's performance), and identifying potential problems that characterize each combination of decision aid function and user supervisory task. We explore the way that training requirements based on the trust model might be used to address these problems. Five training requirements are identified: Acquiring mental models of aid strengths and weaknesses, acquiring decision aid-driven situation awareness and monitoring skills, learning critical thinking skills for novel situations, developing more accurate probabilistic assessments of expected aid performance, and acquiring effective strategies for interacting with an aid as a function of different conditions.

The training requirements framework was tested by employing it to design a training strategy for the Combat Battle Position Recommendation (CBPR) module of the Rotorcraft Pilot's Associate (Chapter 0 and Appendix C). The training strategy addresses three of the five training requirements emphasized in the training framework: It conveys a mental model of features that are correlated with good and bad aid performance, it provides practice in estimating the probability of successful aid performance, and it introduces users to a number of specific user-aid interaction strategies that are appropriate under different circumstances. The illustrative training package was evaluated by four experienced pilots, and received generally favorable evaluations. In their comments, the pilots emphasized two results of the training: acquiring increased understanding of the CBPR aid and learning new ways to interact with it. These findings lay the groundwork for more extensive RPA training development in Phase II of the project. (A step in that direction has been taken by initiating an analysis of the effects of RPA's intent-inferencing capabilities on user trust. This very preliminary analysis is reported in Appendix D.)

The third objective is addressed in Chapter 0, where measures of situation awareness and workload are developed. Both of these factors are significantly affected by decision aid use, but not in a simple way. The presence of an aid, for example, might be expected to increase the user's situation awareness in tasks where the user retains the chief decision making responsibility, by supplying the information needed at the time the user needs it for decisions. On the other hand, to the extent that decision making tasks are handed over to the aid, the need for user situation awareness may actually decrease. As a result, if the user subsequently takes over from the aid, performance may be degraded (Endsley, 1996). Moreover, simply in order to manage the decision aid (including, for example, selecting the appropriate automation mode and monitoring the aid's performance), the user may need to keep track of a whole new set of situation features *to reduce uncertainty about the aid itself*. Appropriate levels of trust in an aid presuppose a new, *decision aid-driven situation awareness*.

Workload has a similarly complex relationship to decision aid use. Traditionally, reduction of workload has been touted as a rationale for the introduction of decision aids. Indeed, perceived workload has been found to play a significant role in users' decisions about when to rely on automation and when to perform a task manually. However, the current consensus is that decision aiding does not reduce the user's overall workload. Instead, it may

permit a more effective allocation of the same total effort, for example, leaving the user free to devote resources to non-automated cognitive tasks. At the same time, however, uncertainty about the aid — and the resulting tasks of monitoring and evaluating its performance —may sometimes *add* to the total workload demands faced by the user.
The ideas in this report are discussed at three levels. Readers who want a quick, non-technical presentation that focuses on pragmatic results can look at the illustrative training package in Appendix C and the discussion of it in Chapter 0. The theory upon which that training was based, including its training implications, is discussed in Chapters 0 through 0, and Chapter 0. Finally, mathematical readers will find the technical basis for many of the theoretical and practical ideas in the footnotes to chapters 2 and 3 and, especially, in Appendix A.

A MODEL OF TRUST IN DECISION AIDS

In important ways training users of decision aids is similar to training any other cognitive skill. The overall task can be broken down into components, which represent the knowledge and skills required and the processes in which they are used. Training can then be designed to convey the required knowledge and skills, through various combinations of instruction, demonstration, practice, and feedback (e.g., Duncan, Rouse, Johnston, Cannon-Bowers, Salas, & Burns, 1996). A description at this level, however, misses some essential elements of training for decision aid use, which make it unique and challenging. We already touched on some of these issues in the Introduction above. Decision aids are tools designed, in part, to reduce demands for situation awareness and workload on the part of human. Yet they impose a set of unfamiliar and demanding new tasks involved in managing the aid itself. To a surprising degree, these unique and challenging aspects of decision aid use hinge on the concept of trust: the degree of confidence the user has in the aid, and the extent to which such confidence is justified and well-informed.
In this chapter, we undertake the task of identifying the requirements that decision aiding systems impose on user training. Our initial goal is a clarification of the concept of trust, and a set of rigorously defined measures for different types of trust. This concept of trust leads directly to a definition of training requirements. Much of the knowledge and many of the assessment skills required for effective decision aid use are directly related to trust. In addition, trust will play a central role in deriving other training requirements to be described in the next chapters.

Previous Work on Trust

Recent research by Muir, Moray, Lee and others has provided a theoretical and empirical beginning for more rigorous study of the role of trust in automation use. Muir (1987, 1994) introduced a multi-dimensional definition of trust (see Table 1); Muir and Moray (1987) described a non-obtrusive method for eliciting subjective assessments of trust from users; Lee and Moray (1994) and Muir and Moray (1996) showed that such assessments of trust could be correlated with subjects' use of automation.
Muir's (1987, 1994) definition of trust in automation borrows from and integrates two models of trust among humans (shown in Table 1). One dimension (from Barber, 1983) specifies three component expectations: *persistence* (of physical, biological, and moral regularities), *technical competence* (at skill-based, rule-based, and/or knowledge-based levels, as defined by Rasmussen, 1983), and *fiduciary responsibility* (the expectation that motives are reliable). The second dimension of Muir's theory (from Rempel, Holmes, & Zanna (1985) is meant to be orthogonal to the first and describes the evolution of trust with experience: from *predictability* (of the machine's behavior), *dependability* (of the machine's enduring dispositions), and *faith* (the conviction that the machine will behave as expected in unknown situations).
Lee and Moray (1992) argued that the two dimensions in Muir's theory were more complementary than orthogonal (as indicated by the arrows in Table 1). They regard faith and fiduciary responsibility as variants of the same concept. They both refer to the basis for trust in situations where the user has little experience with the automation and must fall back on expectations of underlying motives and intentions (of the designer). Similarly, Lee and Moray merge the concepts of predictability and technical competence, claiming that each refers to "stable and desirable behavior or performance." Finally, Lee and Moray map Muir's conceptualization onto a classification of aspects of trust by Zuboff (1988). Zuboff's *trial-and-error experience* is equated to predictability. Zuboff's *understanding* is equated to dependability (and technical competence). Zuboff's *leap of faith* is equated to faith (and hence, to fiduciary responsibility).
We think that there are important distinctions among the concepts that Lee and Moray equate. Unfortunately, Muir has perhaps not adequately defined or clarified these distinctions. For example, it is not at all clear that this is, or is meant to be, a true taxonomy. Persistence, competence, and fiduciary responsibility, for example, do not seem to provide an exhaustive and mutually exclusive classification of "types of expectation." Another problem with Muir's framework, not noted by Lee and Moray, is that consistency and desirability in system behavior are confounded. The value-laden terminology (dependability, faith) seems to preclude the possibility that *distrust* might also evolve as a user acquires experience with a system, or (more importantly) that conditions of trust and distrust might become

better differentiated. (This process of differentiation will be an important topic in our discussion.) A further issue is the relationship between trust in a system as an intervening variable and predictions of system performance. In Muir's framework (see Figure 3 in Muir, 1994), there is a one-to-one relationship between trust and expected automation performance. Thus, we expect that nothing is lost, and some clarity will be gained, by regarding trust as in essence a prediction about the quality of system performance.

The model of trust that we will describe addresses these, and other problems. It depends on two key insights: (1) The qualitative structure of trust is nicely represented by a template for a specific type of *arguments*. Such arguments marshal observations and prior beliefs to make predictions about the quality of system performance over a specified period of time. (2) The quantitative aspect of trust can be conveniently represented by probability distributions over the appropriateness of system actions conditional on features of the system and of the situation: e.g., *p(correct action | system, situation)*. As we shall see, interpreting trust in terms of these two aspects leads to a theory that is more parsimonious, more understandable, and potentially more useful.

Table 1. The first two columns represent Muir's two-dimensional model of trust. Arrows link concepts in Muir's and Zuboff's models that Lee & Moray believe to be equivalent. Numbers in parentheses show the relationship of these concepts to the elements of the APT model described in Figure 2.

Types of expectation (Barber)	**Basis of expectation (Rempel et al.)**	**Aspects of trust (Zuboff)**
Persistence *(3)* Physical Biological Social	Predictability (of acts) *(4)*	Trial & error experience *(2)*
Competence *(1)* Skill-based Rule-based Knowledge-based	Dependability (of dispositions) *(4)*	Understanding *(2)*
Fiduciary responsibility *(2)*	Faith *(4)*	Leap of faith *(2)*

What is Trust? A Model

As noted, our model of trust in decision aids has both a quantitative and a qualitative aspect. The quantitative aspect is provided by the use of probability, and the associated calculus for combining probabilities, as the basis for assessing trust.[1] The qualitative aspect, however, is equally, if not more, important. It provides a structure in which the reasons for (or against) trust can be identified, as well as the sources of those reasons. Because of this duality, we refer to this framework as the *Argument-based Probabilistic Trust* (APT) model.

The qualitative aspect of trust is based on Toulmin's (1958) theory of argument. Toulmin's professed goal was to turn away from the highly abstract character of traditional logic, to examine actual methods of reasoning in real-world domains such as law and medicine, and to develop a theory of logic capable of capturing the rich variety of methods in everyday use. We think some of the same categories apply to reasoning about the expected quality of a decision aid's performance.

The basic framework of an argument, according to Toulmin is shown in Figure 1. A *Claim* is any conclusion whose merits we are seeking to establish. The Claim is supported by *Grounds*, or evidence. The reason that this particular evidence supports this particular conclusion is the existence of a *Warrant*, i.e., a belief in a general connection

[1] As will become clear in what follows, our use of the concept of probability does not imply that probabilistic models are an appropriate description of human decision making, or even that such models should be used as a normative standard for human decision making (Cohen 1993). The usefulness of probabilities in the present context stems from several related sources. They are part of a qualitative representation of the knowledge underlying judgments of trust. As such, they can be used for generating representative sets of scenarios in which users can practice recognizing patterns relevant to trust. In addition, probabilities can be used as *labels* in pattern recognition training, which induce a more refined discrimination among such patterns than non-numerical labels. Similarly, they can be used for generating more refined feedback for pattern recognition performance. Finally, the concept of probability can sharpen users' understanding of decision aid fallibility and clarify the concept of how trust in a decision aid evolves over time.

between this type of Grounds and this type of conclusion, or Claim. The *Backing* provides an explanation of the Warrant, i.e., a theoretical or empirical basis for the existence of a connection between Grounds and Claim. *Modal qualifiers* (such as *probably, possibly, almost certainly*) weaken or strengthen the validity of the Claim. Possible *Rebuttals* are factors capable of deactivating the link between Grounds and Claim, by asserting conditions under which the Warrant would be invalid.

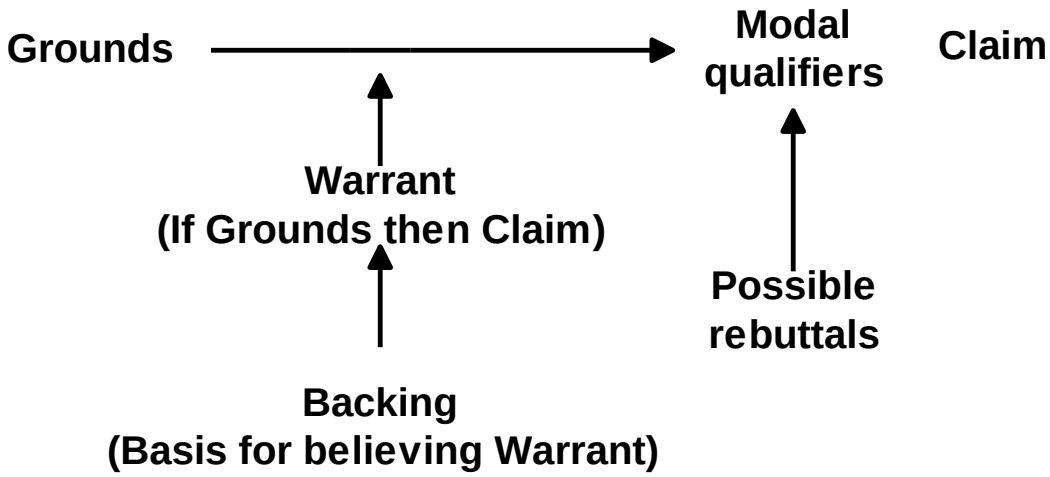

Figure 1. Toulmin's model of argument. The structure can be read: Grounds, so Qualified Claim, unless Rebuttal, since Warrant, on account of Backing.

Figure 2 shows how APT uses the components of Toulmin's model. Table 2 contains examples of arguments for various trust assessment at different phases in the use of a decision aid. We will describe each of the elements of APT in turn:

1. *Warrant:* A Warrant is a general statement that specified conditions are thought to be associated with a specified quality of aid performance. The conditions, which the user believes to be correlated with aid performance, may be features or combinations of features of the system, situation, task, or even specific aid conclusions. The quality of performance may be described specifically (e.g., the system will be wrong under these conditions), probabilistically (e.g., the system is right about 3 times out of 4), or with more vague qualifiers (e.g., the system is highly reliable). The fifth column of Table 2 includes examples of Warrants that draw on different types of features to predict system performance.

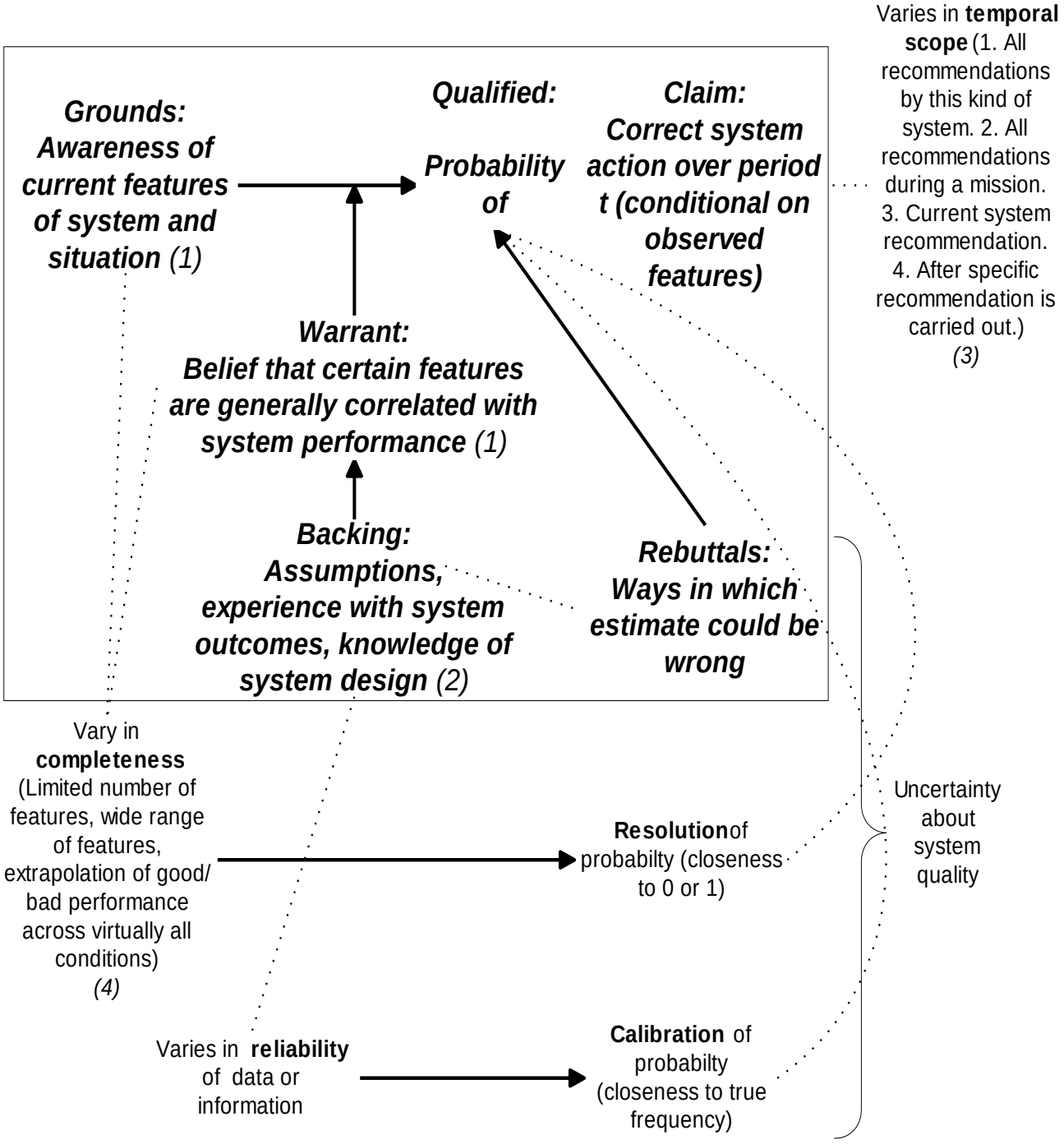

Figure 2. Main components of Argument-based Probabilistic Trust (APT) model. Elements of the argument structure are shown in the box. Parameters associated with these components are shown outside the box, linked to the relevant components by dotted lines. The italicized numbers in parentheses indicate correspondence to the elements of previous models of trust that are shown in Table 1.

2. Grounds: To play a role in an argument for (or against) trusting an aid, a feature or combination of features must not only (a) be believed to affect aid performance in general (as reflected in the Warrant), but must also (b) be observed on the particular occasion for which trust is being assessed (as reflected in the Grounds). Grounds are simply the accumulated observations that influence a user's judgment of the reliability of an aid. The second column of Table 2 includes appropriate Grounds for our set of illustrative arguments.

3. Qualified Claim: The output of the trust model is a qualified claim. This Qualified Claim represents the degree of trust in the system under the conditions specified in the Grounds, according to the Warrant. As noted above, qualifications may be precise (e.g., 30%) or, as is more often the case, vague (e.g., very reliable). The third column of Table 2 shows the Qualified Claims for the set of examples we are considering.

For convenience we will assume that Qualified Claims are always expressed as precise probabilities. Examining this precise sense will be the most efficient way to gain insights into the more vague qualifiers, which may be understood as ranges of possible probabilities (e.g., the chance of being correct is somewhere between .6 and .9). Degree of trust in a system, then, is the probability that the system will produce correct actions over a given period of time, conditional on the grounds, i.e., the relevant features of the system, current situation, task, and/or conclusion.

4. Backing: The fourth component of the model is the origin of the user's predictions about system performance, i.e., how the Warrants were learned or inferred. Users can learn about a system, and develop an appreciation of factors that influence trust, in many different ways: by direct experience with the system, by learning about system design, by talking to more experienced users, or by making assumptions (e.g., best case or worst case). In the above examples, backing consists of the items shown in the sixth column of Table 2.

5. *Rebuttals:* Rebuttals are conditions that potentially invalidate the connection between Grounds and Claim, or conclusion. Rebuttals represent possible exceptions to the rule expressed by the Warrant. They may represent implicit assumptions in the Backing, for example, that one's past experience with the aid has been representative of present conditions. Sometimes, rebuttals reflect explicit assumptions, for example, a decision by the user to assume worst-case conditions for aid validity until he or she learns otherwise. Assumptions are natural and inevitable, since it is impossible to verify every condition that could potentially affect an aid's performance. As a result, any assessment of trust is subject to Rebuttals, even assessments that are based on long experience with the aid or on thorough design knowledge. When events violate expectations, however, assumptions are worth ferreting out and re-examining through a process of critical thinking. The fourth column of Table 2 contains examples of some rebuttals to the claims made in each of the arguments above.

Table 2. Examples of arguments. Each row is an argument regarding system performance, based on different types of features.

Features Used in Argument	Grounds	Qualified Claim (Trust)	Rebuttals	Warrant	Backing
System	This aid is an expert system	so, this system is not very reliable	unless the situation matches very closely one in which the aid was tested, or unless knowledge engineering technology has improved significantly since the experiences I heard about	because expert systems are not very reliable	on account of disappointing performances by many expert systems that I have read and heard about.
System & situation	Dust and leaves are present in about 30% of a given area	so, any battle position recommendation has about a 70% chance of being acceptable	unless there has been a recent heavy snowfall (increasing rotorwash), or heavy rains (reducing rotorwash due to dust), or system modifications to include rotorwash	because if rotorwash is a factor in about X% of an area, this attack-helicopter battle positions aid is (1-X)% likely to recommend positions that do not have rotorwash problems	on account of my experience with the aid in exercises in which I have inferred that the aid does not use rotorwash as an evaluation factor.
System & task	This patient probably has an infectious disease	so, any diagnosis of this patient has about a 95% chance of being correct	unless this is an infectious disease newly introduced to the population, or unless this is one of the cases where one disease closely resembles another	because this particular medical expert system is about 95% reliable in diagnosis of infectious diseases.	on account of my knowledge that more time was spent building the infectious disease knowledge base than any other, and that a goal of 95% correct was set for the project.
System, situation, task, & specific conclusion	The aid has identified a track flying an unusual course as a foe	so, this track has about a 70% chance of being a foe	unless a visual ID of the track as a foe has been obtained	because this situation assessment aid is only about 70% correct when it identifies an unusual track as foe	on account of my observations over many exercises with the aid, that it tends to classify friends as foes if track kinematics are unusual.

Now, let us consider how all these parts are related to one another in complete arguments (i.e., the rows in Table 2). The example in the third row of Table 2 is drawn from the application domain that was the focus of the present research. It represents an Army helicopter pilot's argument about trust in a decision aid for planning an attack. This decision aid — the Combat Battle Position Recommendation (CBPR) module of the Rotorcraft Pilot's Associate — evaluates potential sites from which an enemy, such as a moving tank column, can be engaged. Sites are evaluated in terms of the concealment provided by the terrain, the distance of the battle position from the target vis-a-vis the helicopter's weapon range, the altitude of the site relative to the target, room to maneuver within the site, and so on. However, some factors that are relevant to the evaluation are not considered by the aid. One of these is rotorwash, which is a cloud of dust, leaves, snow or water that may be thrown up by the helicopter's blades, and which can give away the helicopter's position to the enemy. It is usually not possible to determine whether rotorwash will be a factor until the pilot has actually visually inspected a prospective battle position. However, the pilot may have some prior notion of the likelihood of rotorwash in the type of terrain where the mission will take place, and this knowledge may influence the pilot's degree of trust in the aid's recommendation. The example in row four of Table 2 can now be summarized as follows: Thirty per cent of the potential engagement area is affected by rotorwash (Grounds), so the chance the system will recommend an appropriate battle position is 70% (Qualified Claim), unless the terrain has been changed in some way recently (e.g., it may have snowed or rained) and unless other factors also affect the aid's accuracy (e.g., the aid omits other variables that may reduce the quality of the recommendation in this environment) (Rebuttals). This argument is based on the pilot's knowledge that the aid's performance is degraded by rotorwash (Warrant), on account of the pilot's long experience with the aid in situations where its omission of rotorwash as a factor could be inferred (Backing).

How Trust Varies

The most important use of APT is to chart how trust varies, from one user to another, from one decision aid to another, and across phases of decision aid use. To track such changes, APT supplies a set of five interrelated parameters to describe any given assessment of trust. These parameters, as shown in Figure 2, are:

- *Temporal scope* of the Qualified Claim. This is the duration of time that the assessment covers. We will distinguish four principal phases in the use of a decision aid, corresponding to decreasing temporal scope: (1) trust in a system generally over all its potential uses, before a specific mission has been assigned or a specific task has been undertaken (as illustrated by the first row of Table 2), (2) trust in the system's capability for a specific mission or task (as illustrated in the second and third rows of Table 2), (3) trust in a specific recommendation that the aid has made, before the recommendation has been verified or implemented (as illustrated in the last row of Table 2), and (4) trust in a specific aid recommendation after it has been verified or implemented and its quality is known.
- *Completeness* of the Grounds and Warrant. Grounds and Warrant can vary in their coverage of the features that potentially affect system performance. Completeness thus reflects the user's understanding of conditions that might affect trust.
- *Resolution* of the Qualified Claim. Resolution is the degree to which the user can discriminate situations in which aid performance is relatively certain, i.e., the closeness of the probabilities to either zero or one.[2] Completeness affects the resolution of the trust assessment. The more information that is used to predict an aid's performance, i.e., the more completeness in the grounds and warrant, the higher the average, or expected, resolution. For example, a decision aid user may believe that an aid

[2] More precisely, to maximize the resolution of an assessment is to minimize a *resolution penalty*. To calculate the penalty, group together all trust assessments by the user that are equal, or within a specified interval of one another (e.g., all assessments of 75% chance of correct performance are grouped together, all assessments of 50% chance of correct performance are grouped together, etc.). Then determine the actual relative frequency of the predicted event (i.e., correct aid performance) corresponding to each group of assessments. When there are two alternatives (e.g., correct versus incorrect aid performance), the resolution score for a given grouping of assessments is simply equal to the product of the actual probability and 1 minus the actual probability for that grouping. It is obvious that this resolution penalty is minimized (greatest resolution) when the actual probability is 1.0 or 0 (penalty = (1.0) (0) = 0). The maximum penalty (least possible resolution) is represented by an actual frequency of 50% (penalty = (.5.) (.5) =.25). Overall resolution is the sum of these products across the groups of assessments, weighted according to the relative number of responses in each group.
This measure is a variant of Murphy's (1973) partition of the Brier scoring rule (see Brown, Kahr, & Peterson, 1974). It can be decomposed further into a measure of the inherent uncertainty in the problem (the overall probability of the event times its complement) minus the decision maker's ability to reduce it by making distinctions (the variance of the actual probabilities associated with different groupings).

tends to be correct in 80% of all the situations in which it is used. But trust assessments will have more resolution if the user can observationally identify specific types of situations where the performance of the aid is better (e.g., 90%) or worse (e.g., 70%) than the overall average

- *Reliability* of the Backing. This is the amount and quality of data or information that underlies the trust assessment. The more experience a user has with the aid and the situation, the more detailed and accurate the design knowledge the user has, or the more robust the assumptions the user makes, the more reliable is the user's source of knowledge, or Backing. For example, a user who is highly familiar with an area may be confident that the percentage of the area affected by rotorwash is between 30% and 32%, while a user who is less familiar with the terrain may know only that the percentage is somewhere between 15% and 45%.
- *Calibration* of the Qualified Claim. Calibration is the correspondence of the probability estimate to the true frequencies of correct system response given the conditions in the Grounds.[3] The reliability of information and reasoning in the Backing determine the calibration of the probability of correct system performance. For example, suppose each of two users assess the probability of an aid's selecting an appropriate battle position as 70%. The user who is familiar with the area is unlikely to be off by more than 1%, while the user who is less familiar with the area may be off by as much as 15%.

We will consider each of these parameters in turn in the remainder of this section.

Temporal Scope. Predictions of system performance differ in generality and temporal scope. One might want to know the probability that the current aid conclusion is correct; one might want to predict the aid's performance over the next hour (e.g., the expected proportion of correct actions); or one might want to predict the performance of the system across its entire operational lifetime. Trust operates at each of these time windows, which correspond to different phases of aid use. Decisions about reliance on decision aids, which are based on trust, are somewhat different in each phase.

Phase 1. At the longest time window, trust by managers and designers in the relevant technology determines which functions will be automated for the lifetime of that version of the aid. Conversely, trust by users in management, in designers, and in the aid itself helps determine whether the aid will be accepted. Trust by users may also influence the motivation that they bring to training with the aid. Just as importantly, and often neglected, trust at this phase can influence trainers and training designers in the aspects of decision aid use that they choose to address during training.

Sheridan (1976, 1988) has argued that degree of trust in the computer (or more precisely, trust in the computer's designers and the technology they employ) should influence the degree to which "intelligent" functions are assigned to it. He and others (e.g., Riley, 1989; Endsley & Kiris, 1994) have provided frameworks delineating how intelligent tasks may be allocated between user and machine and the different patterns of initiative, permission, and override that may be enforced. For example, according to Riley, a "servant" machine merely processes information and implements actions while the human generates options and makes a choice. Higher levels of automation involve computer generation of options and user selection; computer recommendation of an option with user final selection; computer selection of the option subject to positive approval from the user (Riley calls such a device an "assistant"); computer selection and implementation of the action unless the user overrides it within a specified time period (Riley calls this an "associate"); computer selection and implementation with the user merely being informed of the choice; informing the user only if the user requests to be informed; and informing the user only if the computer decides it is appropriate to do so. At the high automation end of this scale, intelligent devices pass from being associates to being partners, supervisors, and finally, "autonomous." Movement along this continuum clearly is matter of trust, in many cases, the relative trust by managers in aid designers and aid technology versus the users. At the high automation end of the continuum, systems fall into the category described by Roth, Bennett, & Woods (1988) as *prostheses*, that is, they are designed to replace humans in the role of problem solver and employ the human, if at all, only as data gatherer or action implementer. Such aids, including many expert systems, have not received the trust or acceptance that has been earned by simpler types of automation. Roth et al. (1987) describe how experienced users attempt to circumvent an expert system for fault diagnosis (e.g., by turning it off, or tailoring data inputs to influence its reasoning) when they distrust its conclusions. According to many researchers, complete trust in fully automated decision making is hardly ever justified. For example, in the expert system study by Roth at al.

[3] To maximize the calibration of an assessment, we minimize a *calibration penalty*. This is equal to the squared difference between the assessed probability and the actual probability associated with a grouping of assessments (see note 2). Calibration is best when the difference is zero (e.g., penalty = $(1-1)^2 = 0$). Calibration is worst when the difference equals 1 (e.g., penalty = $(1-0)^2 = 1$). Overall calibration is the sum of these squared differences across the groupings of assessments, weighted according to the relative number of responses in each group.

(1987), situations unanticipated by system designers were the "norm rather than the exception." Reason (1990) points out that humans are included in hazardous systems because of their ability to carry out "on-line" problem-solving in novel situations. System accidents, which call for user intervention, are often the result of multiple small failures in combination with latent design defects, whose co-occurrence could not have been foreseen and which, therefore, are truly novel (Perrow, 1984). More generally, it is unlikely that a system designer / domain expert will anticipate all possible contingencies in complex, open-ended, ill-structured domains such as physics, medicine, or combat.

These considerations have important implications for training decision aid users. When an aid's capabilities are oversold, disillusioned users may reject the aid after a single unexpected error. By contrast, users may be more likely to accept an aid when its potential shortcomings as well as its potential contributions are acknowledged in training, and when the need for users to understand both the aid's strengths and weaknesses is emphasized.

Phase 2. At an intermediate time window, for example, during the planning or early in the implementation of an actual mission or task, trust can determine the degree of automation and mode of interaction between user and decision aid. Alternative paradigms for decision aid design have attempted to support the human's role in decision making, by conceptualizing aids as *tools* capable of being adapted in a variety of ways by humans who retain the primary problem-solving role (Roth et al., 1988; Cohen, Laskey, & Tolcott, 1986). In such *human-centered* approaches to design, the user selects the mode of automation (Billings & Woods, 1994). The user can specify the pattern of initiative, permission, and override along a continuum like the one above, treating the aid sometimes as servant, assistant, or associate (and even on occasion allowing it to function autonomously). When an aid is *adaptable* in this manner, the responsibility to assess trust in the computer has effectively been transferred from the designer and management to the user. In the *adaptive aiding* paradigm, by contrast, the computer supports or replaces the user to some degree in the determination of automation mode. The aid in effect tries to predict the appropriate degree of trust in itself (and in the user), and to automatically adjust the sharing of tasks between itself and the human under varying conditions, such as user workload (Rouse, 1988; Andes & Rouse, 1992). When reasoning and decision making are only partially automated, the issue of trust is central not simply to the design and overall acceptance of the aid, but to *the way the aid is used*.

Phase 3. At a shorter time window, during decision making about a particular action, the user's trust in specific aid recommendations determines from moment to moment whether the user will comply with an aid conclusion (Muir, 1988). The influence of trust at this level plays no role in aids that act autonomously. It applies only for aids (or automation modes) that are acting as assistants or associates, i.e., for automation that falls between mere data gathering and autonomous action.

Phase 4. The shortest time window of all involves trust in an aid recommendation that has been directly verified — either by carrying it out, or by observation of the conditions that would determine, or strongly influence, its success or failure. For example, a helicopter pilot may examine the site recommended by a battle position selection aid to determine its suitability. At this final phase of aid use, the user has typically become certain regarding the quality of the aid's recommendation. The probability of a correct response by the aid has become either 1.0 or 0. Information gathered at this stage serves as feedback regarding the accuracy of predictions made at earlier phases, and can shape trust assessments when those conditions recur in the future.[4]

As Table 2 suggests, different *mental models*, i.e., sets of causally relevant features of the system, situation, task, and conclusion, are relevant at these different phases. Training requirements will differ to the extent that trust must be assessed and acted on at different levels of detail and different temporal scope. Thus, training designers must take each phase of trust into account.

Completeness / Resolution. Predictions of system performance are expected to improve with experience. The metric we propose has two dimensions: completeness and reliability. Of the two, we will find completeness to be the more important. It represents users' ability to discriminate among situations that are relevantly different for aid performance. Completeness (and its associated measure, resolution) thus refers at the most basic level to *pattern*

[4] The temporal scope parameter in fact has both a beginning and an end. We have focused on the beginning, i.e., the time *at* which trust is estimated (e.g., during aid design, mission planning, a particular task, or task execution). However, the time *for* which trust is assessed is also important. For example, we construe trust in the CBPR aid as the chance of an acceptable battle position, rather than the chance of a successful battle. A battle may be won from a poor battle position or lost from a good one. We evaluate a Combat Battle Position Recommendation aid in terms of what it is designed to do – select acceptable battle positions – rather than over a more extended temporal scope, in terms of chance of successful battle. Nevertheless, the contribution of battle position to success in battle can be quantified by comparing the chance of a successful battle conditional on a good battle position and the chance of a successful battle conditional on a poor battle position.

recognition. By contrast, reliability (and its associated measure, calibration) refers to the numerical labels (i.e., probability assessments) that are attached to the discriminated situations.

Completeness is the percentage of the relevant features that are accounted for in the user's judgment. To be relevant, a variation in conditions must plausibly threaten to degrade the system's performance. For example, one might ask how well a situation assessment aid will hold up as the number of targets is increased, how well an expert system performs in knowledge-based (or novel) tasks as opposed to rule-based (or routine) tasks, or how well a planning aid performs on problems in which tradeoffs among goals must be handled as opposed to problems in which there is only a single goal. Completeness refers to the proportion of relevant variations in conditions under which the aid's performance has been experienced or inferred and which enter into the Grounds of the present prediction. The more such variations in conditions that have been tested, the more complete is the user's understanding of the aid. Testing can take place via any appropriate knowledge source or strategy, which serves as Backing for the judgment; for example, by actual experience of the aid under multiple variations of the relevant conditions, by inference from knowledge of the design of the aid, or even by assuming that the aid is like a human in its abilities. Strategies like "fiduciary responsibility," worst-case assumptions, or projection of animistic traits can also provide a basis for differentiation of conditions, although arguments based on such assumptions are vulnerable to numerous rebuttals.

One result of increasing completeness is improved *resolution* in predictions of aid performance. For example, a user may begin with a single undifferentiated prediction of aid performance: *p(correct action | system)* = .6. With experience, the user discovers that the aid performs better in routine conditions and worse in novel conditions. After this experience, the new predictions become: *p(correct action | system, situation = routine)* = .8, and *p(correct action | system, situation = novel)* = .4. If the actual probabilities of correct aid performance are different in these two conditions (even if the assessments of .6 and .4 are not right), then the user's resolution has increased (see note 2 above). Note that the user must monitor the situation for routineness in order to make use of the knowledge gained from experience. If the user is not aware whether the current situation is routine or non-routine, the user must fall back on the original, less differentiated prediction. Thus, resolution depends both on stored knowledge and on dynamic situation awareness. In general, higher resolution means that probabilities will tend to be closer to the extremes of 1.0 and 0. In other words, as the user *learns about* and *observes* a larger proportion of the relevant variables, uncertainty is reduced about whether or not the aid will produce a correct response.

Reliability / Calibration. The second dimension of improvement with experience is reliability. This refers to the amount and quality of data or inferential credibility underlying a particular probability. For example, if the user has operated an aid only a few times in non-routine conditions, or if he has only a poor understanding of its design, reliability of his prediction may be low: e.g., *p(correct action | system, situation = novel)* = .4 ± .2, where "± .2" represents a rather large interval of uncertainty. After more extensive experience (or better understanding of the aid's design), the reliability of his prediction may be much higher: e.g., *p(correct action | system, situation = novel)* = .45 ± .03.

It might be expected that direct experience with an aid would produce more reliable estimates of performance than design knowledge. But this is not necessarily the case. Each of these sources of knowledge can provide a check on the other. An understanding of the aid's design, for example, might lead a new user to overlook early chance failures of the aid that are not truly representative of its capabilities. As another example, design knowledge may help a relatively experienced user realize that the aid's performance under one set of circumstances (e.g., for which it was not optimized) is not predictive of its performance in some other set of environments. Conversely, a conflict between users' experiences with the aid and predictions based on design specifications may be an important clue that assumptions made by aid designers are mistaken. In general, conflict among different sources of knowledge is a symptom of systematic error or wrong assumptions, and, if responded to appropriately, can add to reliability (Cohen, 1986). Training that includes alertness to such signs of trouble may thus improve the appropriateness of trust and the effectiveness of decision aid use. More reliable estimates, whether based on experience or design knowledge or both, are more stable and thus less likely to be influenced by exceptional experiences.

As reliability increases, the *calibration* of predictions should also improve, both through reduction in random error (e.g., due to sampling or measurement) and reduction in systematic bias. Calibration means that the user's estimated probabilities correspond to the actual probabilities. For example, whenever the user assigns a probability of .6 to an event, the event actually occurs 60% of the time (see note 3 above). In that case, the user is well calibrated.

Resolution and calibration are orthogonal dimensions of trust assessments. As we have noticed, resolution increases when the user's assessments of trust distinguish situations that really differ in terms of aid performance. Calibration is an added feature, in which the user's assessments actually match the true probabilities of aid performance in the situations that the user distinguishes. Just as a user might distinguish many truly different situations without getting the actual probabilities right (high resolution, low calibration), so might a user get the probabilities right without distinguishing many different situations. The first is usually a more desirable state than the second. The classic

example of high calibration and low resolution is the weatherman who forecasts the probability of rain for the next day by looking up precipitation frequencies for the entire year. He is well calibrated, but not very informative. Calibration and resolution of probability estimates tend to trade off with one another (Brown, Kahr, & Peterson, 1974). Higher resolution demands that we divide up data into more numerous (but less well sampled) categories, or that we make a larger number of (possibly unreliable) inferences. In our earlier example, the low-resolution assessment — *p(correct action | system)* = .6 — is based on all the data, and thus will be more reliable, than the high-resolution assessments — *p(correct action | system, situation = routine)* = .8, and *p(correct action | system, situation = novel)* = .4 — which are each based on only a portion of the data.

Users may improve resolution by making assumptions. As indicated above, there is a tradeoff: The higher-resolution judgments will be subject to rebuttals, corresponding to potential failure of the assumptions. For example, suppose that a particular feature of the situation is highly correlated with aid performance. In particular, if the value of the feature is greater than x, trust in the aid is 20%, but if the value is under x, trust is 98%. Suppose further that quite different strategies for using an aid are appropriate in these two situations, and a decision aid user does not know what the actual value of the feature is. In planning, it will not be very useful to take an average, or expected value, of the two possibilities, e.g., trust of 59%. This is an abstraction that does not correspond to any of the actually possible situations, and there is no way to plan for it. It lacks resolution, since it fails to discriminate the two situations. Although 59% may be well calibrated with respect to the more general, "average" situation, it will not be very close to either of the two probabilities that may actually be realized. The user may improve resolution, and have a fair chance to achieve calibration, by adopting the worst-case assumption. At least this user will be well prepared for one of the two actual situations.

Assumptions, of course, can also play less defensible roles in trust assessment. Assumptions may be carefully and explicitly adopted; alternatively, they may be implicit in the user's habitual responses to an aid. For example, by projecting animate features into the aid, a user might feel quite confident that the aid will have trouble in the same situations that the user finds difficult. The apparent increases in resolution that assumptions bring must be purchased at the cost of dependence on assumptions, which may themselves be unreliable. Training in critical thinking skills may help users detect their assumptions and stay alert to signs that assumptions are wrong.

Resolution, calibration, and potential rebuttals, taken together, describe the uncertainty of a user's judgments about how much he or she should trust a system.

Event Trees: Trust as an Unfolding of Events

A key feature of trust is that it evolves as the user gains experience with an aid, and as the user moves through the various phases of a particular mission or task. In this section, we delve more deeply into this important aspect of APT. In so doing, we also explore and clarify the elements of the APT model, as well as some important relationships among its parameters. As always, the quantitative aspect of the model, i.e., probabilities, are secondary to the qualitative aspects, and are of value primarily for the light they shed on qualitative relationships.

A key tool for understanding how trust evolves over time is the concept of an *event tree* (Shafer, 1996). An event tree can be thought of as a succession of observations or experiences, in each of which the decision-aid user learns something new that is potentially relevant to predictions about aid performance. The outcome of all these observations and experiences determines a particular path through the tree. Each such path represents a possible story, or scenario, about decision aid use. The end of each story is either a successful or unsuccessful contribution by the decision aid to the user's task (e.g., a successful attack). The components of our trust model, as well as their parameters, can all be defined with respect to such event trees. The event tree itself can be regarded as a summary or encapsulation of the user's *mental model* of the decision aid's performance. It includes the factors that the user believes are relevant and their implications for predictions of aid accuracy.

In our discussion, we will draw on examples from a Combat Battle Position Recommendation (CBPR) aid such as the one being developed for the Rotorcraft Pilot's Associate program. The examples, however, will be highly simplified in order to make the relevant principles as clear as possible.

Event Trees for Decision Aid Use

We have seen that a user's interaction with a decision aid can be divided into major phases:

1. Prior to being assigned a mission in a particular geographical area, the user may train with the aid for a variety of different missions under different conditions of terrain and enemy tactics.
2. After being assigned a mission, the user makes choices about how he or she will interact with the aid in the given situation or task. For example, during mission planning or early in the execution of a task, the user makes decisions about automation modes.
3. Later in the execution of the mission, after the aid makes a specific recommendation, the user may (if the selected automation mode allows the user to monitor the aid) make a decision regarding acceptance, rejection, or further verification of the aid's recommendation.

4. After the user has verified or implemented the aid's recommendation, most if not all of the uncertainty regarding the accuracy of the recommendation is resolved.

With successive phases of use, more and more information about the performance of an aid is available. Before being assigned to a mission, the user may have knowledge regarding the types of situations and tasks the aid can address and a rough sense of its likely accuracy in each one. After mission planning begins, there will be more specific knowledge about factors, such as terrain and enemy tactics, that will influence the performance of the aid. Finally, during the execution of the mission, the user will have even more information, such as actual inspection of the terrain, by means of which to evaluate a specific aid conclusion.

This process of acquiring information over time can be conveniently visualized as a progression along the branches of an *event tree* (Shafer, 1996). The event tree represents all the factors that are known to affect the aid's accuracy, organized in the sequence in which they are expected to be observed by an aid user. Before the user is assigned to a theater of operations (Phase 1), the user can be thought of as implicitly facing a set of branching possibilities for how and where the aid will be used. Later, when a mission is assigned and a task is undertaken (Phase 2), the user will have actually traveled along one of the branches that faced him or her earlier as a possibility. At the mission stage, too, the user faces a set of branching possibilities, representing uncertainty about what the aid will actually recommend. As the mission is carried out and the aid generates specific recommendations (Phase 3), the user will travel along one of these branches. Even after the aid's recommendation is known, uncertainty about the outcome may not be completely resolved until the user actually tries out the aid's recommendation (Phase 4).

Figure 3 is an event tree which illustrates the kind of information that might become available to a user at each phase of use. This user is aware of two factors that bear on the accuracy of the Combat Battle Position Recommendation aid. First, the aid does not take into account the angle of attack against the enemy in its evaluation of candidate battle positions. However, a frontal assault has less chance of success than a flanking or rear attack. Second, the aid does not take rotorwash into account in its evaluation. Yet dust, leaves, snow, or water blown upward by the helicopter's rotors can give away its position, reducing the chances of a successful attack. The user believes that terrain interacts with both of these factors. For example, the user believes that rotorwash is worse in the desert than in the mountains, and that a rear or flanking battle position is more likely to be selected by chance in the desert than in the mountains. In this example, the only definite information the user has prior to assignment of a mission (in Phase 1) is the nature of the system itself, represented by the arrow on the far left. Once the user is assigned a mission (Phase 2), however, he learns whether the aid will be used in desert or mountain terrain. After the mission is underway, in either of these two types of terrain, the aid will actually recommend a battle position (Phase 3), and the user can draw on his or her own judgment of likely enemy avenues of approach to determine the angle of attack implied by the recommendation. On the other hand, it is often impossible for pilots to determine if rotorwash will be a problem until they visually inspect a candidate battle position (Phase 4). In a real example, of course, a user's event tree might be more complicated, since the user might make many more relevant observations at each stage.

An event tree can be used to specify the probabilities underlying a user's trust in a decision aid. In Figure 4, we have assigned probabilities summing to 1.0 to the branches emerging from each node. These reflect the chance that the factor corresponding to each branch will in fact occur. For example, in the pre-mission stage (Phase 1), the user believes there is a 60% chance of being assigned to a desert region, and a 40% chance of being assigned to a mountainous region. If the mission turns out to be in desert terrain, the user believes the chance is three out of four that the aid's recommended battle position will be a rear or flanking one rather than frontal. But if the mission turns out to be in mountainous terrain, the user believes the chances are even. Finally, if the mission is in the desert, the user believes that about 30% of the terrain will be subject to rotorwash; therefore, there is approximately a 70% chance that the aid's recommended battle position will be acceptable in this respect. On the other hand, if the mission is in the mountain, the user believes that rotorwash is a factor in only about 5% of the potential sites; thus, the aid's chance of recommending a suitable battle position in this respect are 95%.

At the end of each path in Figure 4 is a 1.0 or 0, representing an acceptable or unacceptable battle position recommendation, respectively. This is not the same as the chance of a successful attack, since an attack might fail for other reasons, even though the aid selected an ideal battle position. (For example, friendly forces might be attacked enroute to the battle position, or the ordnance employed at the battle position might be inappropriate for heavily armored targets.) For an attack to succeed, the battle position must be appropriate *and* other factors must also be right. The numbers at the far right nodes of this tree thus stand for the chance that the attack will be *successful as far as battle position selection is concerned*, i.e., that battle position will not be a cause of failure of the attack.

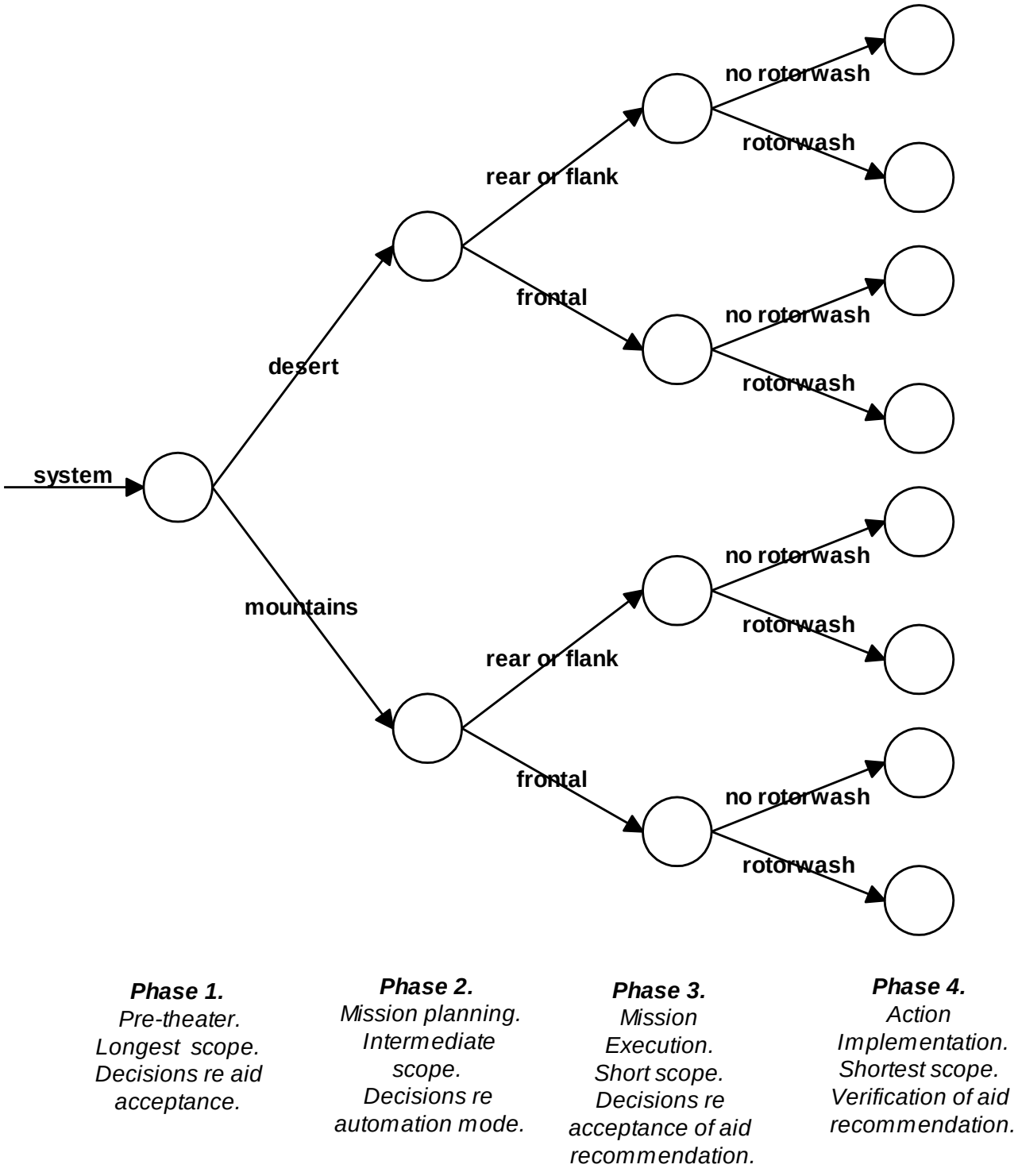

Figure 3. An event tree in which relevant information is acquired at every phase of decision-aid use.

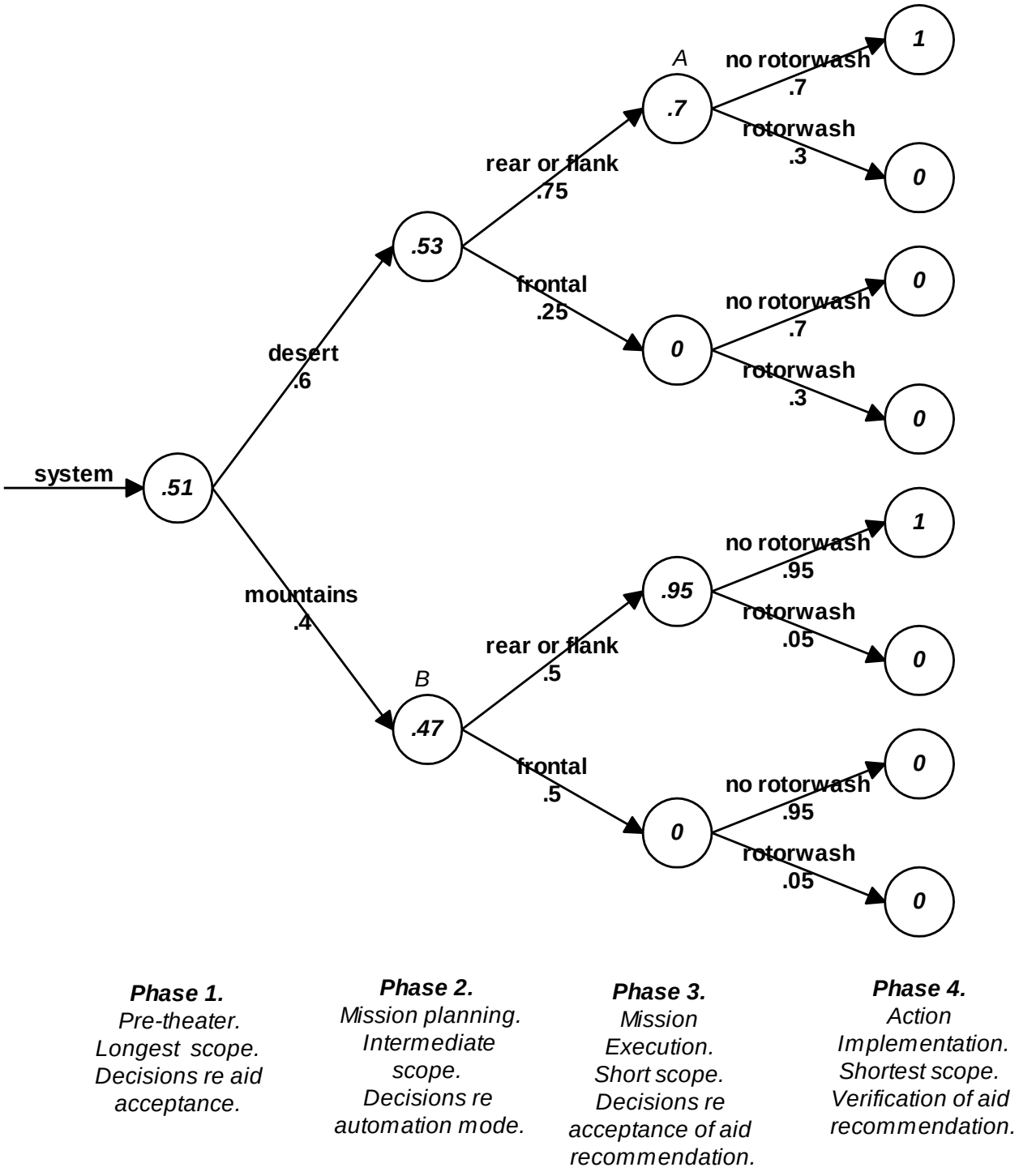

Figure 4. An event tree with illustrative probabilities for each branch.

The numbers within the circular nodes of Figure 4 represent the trust the user has in the aid when arriving at that point in the tree. Trust is the expectation of the user at that point that the aid will take one of the paths that end with a 1.0 rather than a 0.

Figure 4 assumes that the presence of rotorwash or a frontal angle of attack renders a battle position completely unacceptable (i.e., implies that the chance of successful attack = 0). A final complication involves the case in which

different features have graded effects on the acceptability of an aid's recommendation. For example, a helicopter battle position where rotorwash is a factor may be somewhat more acceptable than one that involves a frontal attack. Suppose that a user regards the probability of a unsuccessful attack due to battle position, given that rotorwash is present, as only 25%, but regards the chance of failure given a frontal angle of attack as 65%. This can be accommodated easily within event trees by expanding each of the terminal nodes at the far right in Figure 4 with additional branches, as illustrated in Figure 5.

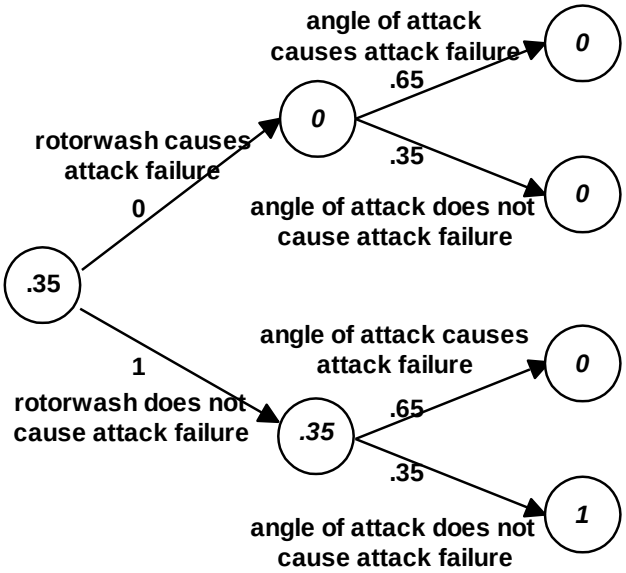

Figure 5. An illustrative subtree that could be tacked onto either of the far right nodes in Figure 6 that involve no rotorwash and frontal angle of attack.

The first additional node represents the chance that rotorwash will or will not be responsible for failure of the attack. In Figure 5 it happens that rotorwash cannot cause the attack to fail, because rotorwash is not present in the part of the tree represented. The second additional node represents the chance that angle of attack will or will not be responsible for failure of attack. In the part of the tree shown in Figure 5, angle of attack is frontal, so the chance that angle of attack will cause failure once the attack is launched is .65. There are four possible combinations, and each ends in zero (representing unsuccessful attack) except the one in which *neither* rotorwash *nor* angle of attack causes failure of the attack, which ends in 1.0. Since the chance of this combination is .35, that is the chance of success associated with these particular conditions, *from the point of view of battle position*. (Of course, the attack could fail for other reasons even when the final probability is one. See footnote 4.)

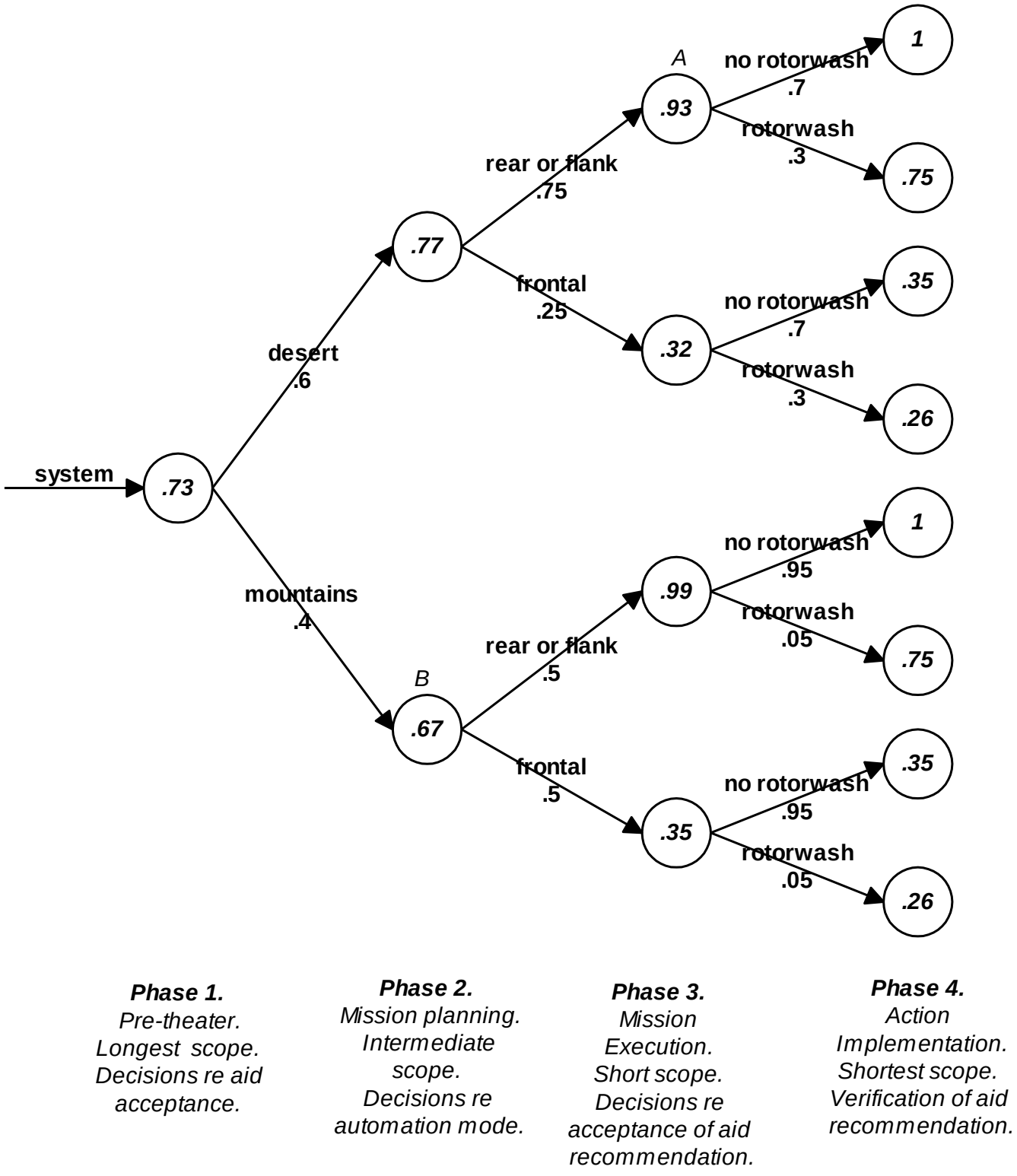

Figure 6. Event tree representing graded effects of rotorwash and angle of attack on chance of success of attack.

These chances of success are easy to calculate without actually drawing all the additional branches. Figure 6 depicts a tree in which the graded effects are represented by probabilities of successful attack placed in the final nodes of Phase 4. It can be seen that including graded effects leads to a significantly higher level of trust in this example.[5]
We are now in a position to clarify APT's components and parameters in terms of event trees with probabilities.

Event Trees and Components of the Trust Model

Warrant. An event tree with probabilities can be thought of as a compendium of all the Warrants a user might draw on for arguments about trust. We can use it to derive generalizations about the connection between potential observations (features of the situation, aid, task, or aid recommendation) and aid performance. For example, suppose the user whose event tree is shown in Figure 6 is in fact assigned to the desert and receives an aid recommendation that involves a rear or flanking attack. This user is at the node labeled *A* in Figure 6. For this user, the major remaining uncertainty about aid performance is due to the possibility of rotorwash. Since the user believes that approximately 30% of this terrain will be affected by rotorwash and that there is a 75% chance of successful attack with rotorwash, the user's trust in the aid (probability of an acceptable recommendation) is (.70) (1.0) + (.30) (.75) = .93. The event tree in Figure 4 thus embeds within itself the following warrant: "If this system is used in the desert and recommends a battle position that involves a rear or flanking attack, the chance of the recommendation's being acceptable is approximately 93%."

This event tree implies a Warrant for trust assessments not only in the situation we just discussed, but at every vantage the point the user might encounter. Each such vantage point, represented by the circular nodes, involves a series of observations corresponding to the branches leading *to* it. And each such vantage point also corresponds to a specific prediction regarding the performance of the aid, which can be calculated from the branches leading *from* it (if there are no branches leading from it, then trust is represented directly by the number within the circle at that node). A Warrant is simply a pairing of a set of potential observations on the path leading up to a node, and a prediction of system performance based on all the paths leading out of that node.

Grounds. The Grounds for a trust assessment consist of the observations that were made by the user as he or she moved along the path to the currently occupied node. Each time the user advances along another branch of the event tree, the observation corresponding to that branch is added to the Grounds for the next trust judgment. Thus, the Grounds for a trust assessment in Phase 1, as shown in the tree of Figure 6, is a knowledge of system features that are common to all its potential uses. By the time the user gets to Phase 2, a mission has been assigned, and the Grounds include system features plus a knowledge of the terrain (desert or mountain). In Phase 3, the system has made some specific recommendation, and Grounds include system, terrain, and knowledge about the recommendation (i.e., that it involves either a rear or flank attack, or a frontal attack). Once the recommendation has been verified by observing the recommended site in Phase 4, the Grounds also include "no rotorwash" or "rotorwash."

Qualified Claim. At each node in the tree, the user can determine the probability of a correct aid conclusion conditional on the Grounds. We have already seen that the user's trust in the aid is 93% at point *A* in Figure 6, after being assigned to the desert and receiving a recommendation that involves a flanking or rear attack.

As noted earlier, numbers within the circular nodes of the tree represent trust at that point in the event tree. Trust is simply the *expected*, or average, aid performance as seen from that particular viewpoint, and it is determined by looking toward the possibilities (if any) branching from the node toward the right. There are, of course, no branches emerging from the nodes on the far right of the tree. The numbers within these nodes reflect the probability that the aid's battle position recommendation will contribute to success, rather than failure, of the attack. The numbers in the terminal nodes in Figure 4 happen to represent certainty regarding the aid's contribution to successful attack, and so they describe probability = 1 or probability = 0 of correct aid performance, respectively. In Figure 5, by contrast, the numbers in the terminal nodes represent various degrees of uncertainty regarding aid performance.

The numbers in the other circular nodes can be calculated easily from these terminal nodes and the branch probabilities. For example, suppose the user is assigned to mountainous terrain, and thus is at the node labeled *B* in Figure 6. How much trust does this user have in the aid at this phase of its use, based on knowledge of the system and terrain? There are four possible pathways leading from node *B* to the situations at the far right. The probability of any one of these pathways is calculated by multiplying together the probabilities on its component legs, including

[5] Other possible complications to the model include, for example, accommodating more than two levels of the features, non-independence of the effects of different features, and variations in the sequence with which features are observed. In addition, it is, of course, likely that more than one feature relevant to decision aid performance would be observed at a single phase. These can all be handled straightforwardly within event trees. Finally, note that all the probabilities in this and subsequent figures are purely illustrative and not meant to represent the true frequencies in any actual combat context.

the probabilities in the relevant terminal nodes. The user's trust at point *B* is then the sum of the probabilities of the pathways leading from *B* to the end of the tree. In this case, trust is (.5) (.95) (1.0) + (.5) (.05) (.75) + (.5) (.95) (.35) + (.5) (.05) (.26) = .67. In other words, trust is the chance that the user, starting at point *B,* will end up in a situation on the right representing a successful contribution of the Combat Battle Position Recommendation aid to the attack.[6]

Backing. Backing refers to the sources of knowledge underlying an event tree. The event tree representation itself suggests an interesting distinction among such knowledge sources. Assessments of trust are always made from the point of view of some node in an event tree. The distinction between *external* and *internal* Backing concerns whether the relevant knowledge pertains to events that came *before* the current node, or to events that are expected to occur *after* the current node, respectively.

Internal Backing generates predictions of paths that could be followed in the future. For example, if a user is at node *B* in Figure 4, the user's trust might be based on expectations regarding the aid's recommendation (either a frontal attack, or a rear or flanking attack), and the site (either be subject to rotorwash or not). In the previous section, we showed how the trust of a decision maker at point *B* can be derived from the probabilities of these future paths.

The knowledge required to generate predictions of future paths can come from many sources: the user's experience with the system in many different environments and tasks, the user's experience with analogous systems, reports by other users of their experiences with the same or similar systems, projection of the user's own strengths and weaknesses into the aid, and/or inference from knowledge of system design. In fact, more than one of these sources might be available to a user simultaneously, e.g., a user who has both design knowledge and personal experience with an aid. This user must then somehow reconcile and fuse the information to produce a trust assessment. Multiple confirming sources of knowledge will increase the *reliability* of the trust assessment, while conflicting sources will reduce it.

On the other hand, assessments of trust can be based on a mental model of events that are *external* to the user's current node in an event tree. In this case, the user's mental model does not differentiate future paths representing possible conditions, or, if it differentiates them, does not assign definite probabilities to them. Rather, it associates a particular level of trust directly with events in the tree that have already happened. For example, a user might have learned to associate particular levels of reliability with all physical systems, all biological systems, or all social systems (Figure 7). This temporal phase comes before the pre-theater phase of decision aid use, since it predates users' awareness that they will be dealing with a decision aid at all. This abstract level of trust corresponds to Muir's notion of the *persistence* dimension. Whether this abstract level of trust plays much of a role, we do not know. More plausibly, however, users may associate a particular level of trust with specific kinds of physical systems, such as decision aids. More specifically still, they may associate a particular level of trust with specific kinds of decision aid, e.g., expert systems versus database systems versus sensor data processing systems. Similarly, suppose the user at node *B* in Figure 6 never learned the importance of angle of attack or rotorwash for the aid's accuracy. However, over many experiences with this particular system in mountain terrain, the user developed a sense of its likely overall accuracy in such environments. The event tree for such a user might be represented by Figure 8, rather than by Figure 6.

Models of external events may come from a variety of sources: the user's own experience with systems, situations, or tasks of the relevant kind; reports of others' experiences; projection of the user's own strengths and weaknesses; and/or knowledge of design limitations. Here, too, multiple sources may provide competing or confirming bases for a trust assessment in an event tree.

In addition to having multiple sources for the probabilities in a given event tree, users may also have more than one event tree for a particular judgment of trust. In fact, users might employ both external and internal models as Backing for a single trust assessment. For example, a user at node *B* in Figure 6 will arrive at a trust assessment of 67% based on internal modeling, i.e., prediction of future paths with respect to angle of attack and rotorwash. But the same user may simultaneously be at node B in Figure 8, having formed an overall impression that the reliability of the aid in mountain terrain is approximately 65%. Moreover, this user may also have general feelings of trust in physical systems, at the 70% level, as represented in Figure 7. This user's trust assessment will be a reconciliation, selection, or fusing of the separate arguments represented by these three event trees.

---

[6] The tree in Figure 6 can be expanded until its terminal nodes also end in 1 or 0, representing a successful or unsuccessful battle position, respectively. The method for expanding the tree is illustrated in Figure 5, and continues the temporal sequence of observations depicted in the rest of the tree. The first branching represents the possibility that rotorwash will cause the attack to fail. The chance of this is zero if there is no rotorwash, and a .25 chance if there is rotorwash. The next branching represents the possibility that angle of attack will cause the attack to fail. The chance of this is zero if angle of attack is rear or flanking, and .65 if frontal.The values within the terminal nodes in Figure 6 represent the chance that the user will end up at a node with a 1 in the expanded tree we have described.

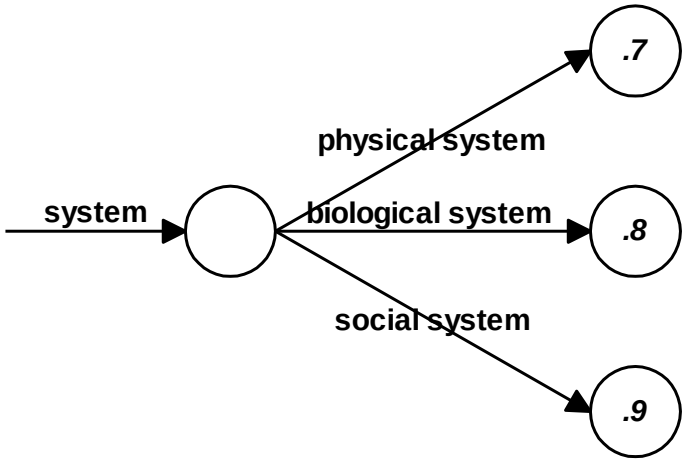

Figure 7. Abstract level of trust assessment, based on systems of different types, prior to an encounter with a specific decision aid. This fits Muir's dimension of *persistence*.

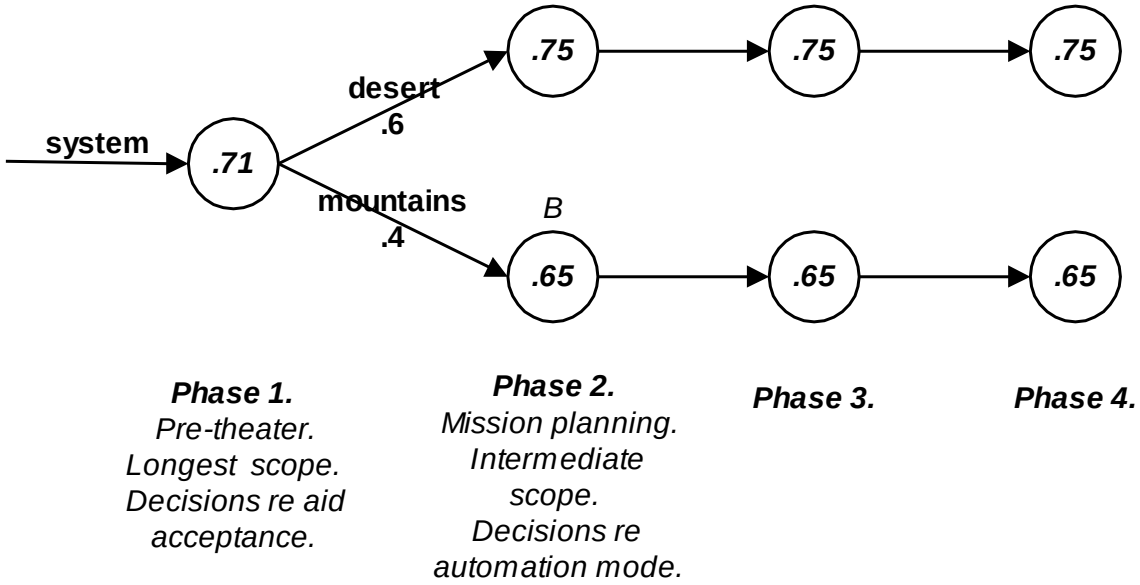

Figure 8. The trust assessment at point B is based on external Backing. Contrast with Figure 4, where a trust assessment at this same point is based on internal Backing. No further information is acquired in Phases 3 and 4, so the trust assessment remains the same.

Rebuttals. Rebuttals refer to assumptions underlying an assessment of trust. Such assumptions may be either implicit or explicit, but they are both necessary and inevitable in the generation of probabilities for an event tree, i.e., in constructing a Warrant for an argument about trust. However, assumptions are by definition open to challenge. If an assumption is rejected, judgments of trust that depended on it may be dramatically changed.

Typically, assumptions arise because of incomplete knowledge of the domain, incomplete experience with an aid, and/or inadequate understanding of the aid's design and its implications. For example, suppose a user of the Combat Battle Position Recommendation aid does not understand the importance of angle of attack in selecting a battle position, or does not realize that the aid omits it. The event tree for this user may look like Figure 9. This tree resembles the tree in Figure 6, except that angle of attack is missing. This tree reflects the users implicit assumption that angle of attack is irrelevant, or, equivalently, the implicit assumption that angle of attack will always turn out to be flanking or rear. For example, the probability of correct system action, given that the aid is used in the desert, is 93%. Notice that 93% is an accurate assessment of trust if we expand the Grounds to include the additional condition that the recommended battle position is on the flank or rear. Similarly, 99% trust is an accurate assessment of trust given that the aid is used in the mountains (point *B*), with the implicit assumption that angle of attack is ideal. As a result of this implicit assumption, the user's trust in the aid is consistently higher than it should be.

Experience might teach the user that his or her assessment of trust was wrong. For example, the user might have the event tree of Figure 8, representing a direct impression of aid performance, as well as the more detailed breakdown in the tree of Figure 9. In particular, experience leads the user to directly associate a probability of correct system response of 75% with desert terrain, and to directly associate a probability of correct response of 65% with mountain terrain. This observed frequency of success is far less than what the user would expect based on the detailed event tree of Figure 9. The two sources of knowledge about system performance conflict with one another.

One option for resolving conflict, though not a very good one, is to average the competing estimates. In this example, the result would be a trust assessment of 84% for the desert and 82% for the mountains. There are at least two problems with this approach: (1) It provides an inconclusive result, i.e., almost identical probabilities of success for desert and mountains, and therefore little guidance for decisions about automation mode that must be made at that phase; and (2) it offers no explanation of the conflict; the user learns nothing new about the system's performance. Another, more fruitful approach is to use the conflict as a symptom that something is wrong in one's mental model of the aid and/or the situation. The solution is to probe deeper for causes of the conflict, by looking for mistaken assumptions underlying one or the other of the conflicting assessments (Cohen, 1986). In this example, the user might realize that the event tree in Figure 9 must be incomplete. There is some factor in addition to rotorwash affecting the system's performance. This realization may initiate a process of more careful monitoring of the aid, which eventually leads both to a more accurate assessment of trust in the present case and to a more accurate mental model for use in the future. The new probabilities of successful aid performance in Figure 6, which includes angle of attack, agree closely with the trust assessment based on direct experience in Figure 8. This confirms the idea that the internal model represented in Figure 6 is relatively complete, at least with respect to the situations encountered so far.

Rebuttals are a reminder that a user's mental model of the aid is never quite finished or perfect. It is always possible for the aid to behave in unexpected ways in new situations, because of some overlooked factor. It is neither possible nor worthwhile to try to enumerate and test all the assumptions underlying a particular assessment of trust. Hidden assumptions are worth ferreting out, however, in situations where the current mental model proves inadequate. As noted earlier, a critical cue for the need to consider rebuttals is conflict among different sources of information. For example, when direct experience seems to conflict with what is understood about system design, either experience has been limited or design understanding is flawed, or both. As another example, when another user's assessments are inconsistent with one's own, there may be non-overlapping experiences or design insights that one or both could share with the other. In all these cases, one or more of the competing mental models of the aid must be mistaken. The lesson for training is important: Users must learn to monitor not simply for the specific features that signal degraded aid performance (as in the event trees we have considered), but should also monitor for more subtle signs of trouble with the event tree itself, such as conflicting assessments of trust.

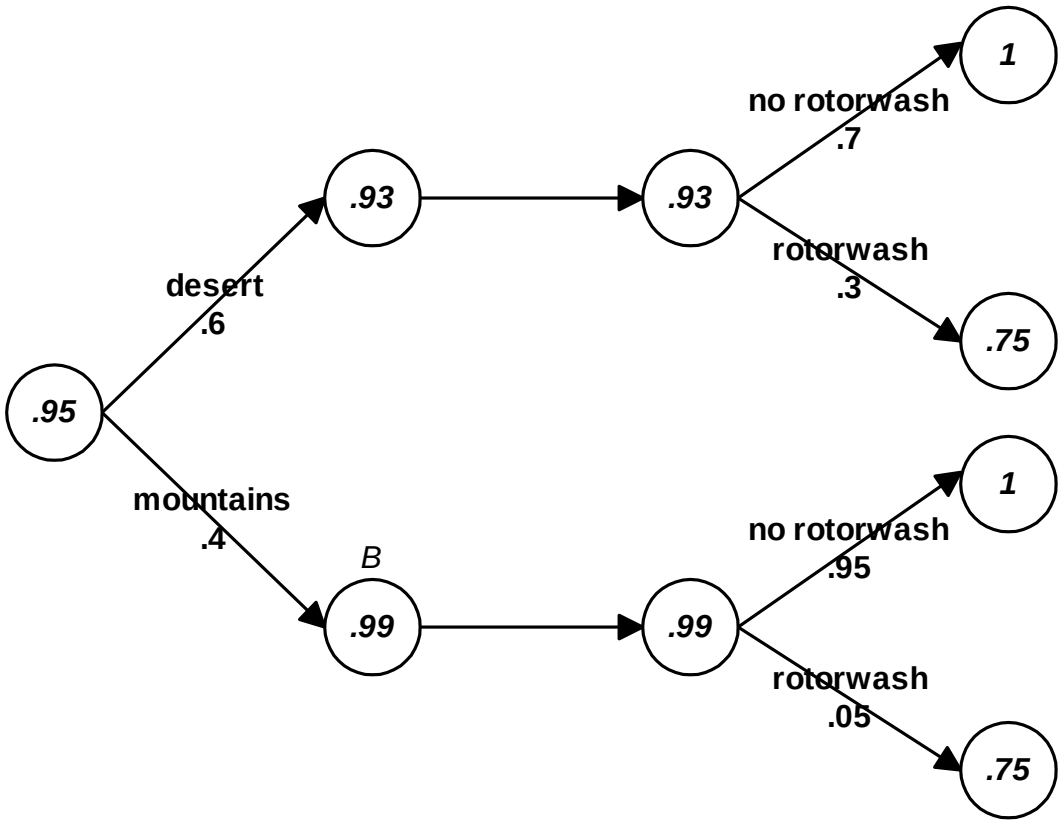

Figure 9 Event tree in which the user is unaware of the importance of angle of attack. As a result, trust is higher than the in the fuller event tree of Figure 6, and no updating occurs in Phase 3.

Event Trees and Parameters of Trust

Event trees also clarify the parameters of the APT model and their interactions.

Temporal scope. The event tree representation makes clear the sense in which larger temporal scope corresponds to more general judgments of trust, i.e., judgments that cover more possible cases of decision aid use. As the user moves along the phases of aid use from left to right, the *temporal scope* of the assessments decreases, from a consideration of the entire event tree at the extreme left node (prior to assignment to a theater), to consideration of more and more restricted sets of possibilities represented by smaller and smaller subtrees, as the user moves into a mission, and from there into a specific task.

Completeness. Completeness in a judgment of trust refers to the coverage of relevant features in both the Warrant and the Grounds. Completeness of the Warrant involves knowledge and experience that are utilized to build richer event trees. Completeness of the Grounds involves dynamically updated situation awareness (Chapter 0) regarding the factors in the event tree, so that judgments of trust remain up to date. The idea that situation awareness must be expanded to include factors that are predictive of decision aid success is an important implication of our model. We will refer to this as *decision-aid driven situation awareness*.

Estimates of completeness obviously require some standard of comparison. How do we know what set of features constitutes a "complete" event tree for a given decision aid? The answer is, of course, that we do not. As we saw in the discussion of Rebuttals, no mental model of aid performance is ever likely to be complete or perfect. However,

aid designers, instructors, and long-time users of the aid (if any) are likely to have the most complete models. These can serve as the standards to which a user's understanding is compared. The standard of completeness for a mental model is simply another mental model — the fullest and most accurate one available.

Incompleteness in a user's event tree always has at least one consequence: inability to update estimates of trust on some occasions when relevant new information is available. Incompleteness may sometimes, but not always, have a second consequence: inaccurate, or uncalibrated, estimates of trust. We will suppose, for the sake of illustration, that Figure 6 is the most complete event tree available, and use it as our standard. Then Figure 8 is an example of an incomplete event tree in which the estimates of trust are approximately correct (the probabilities of correct system performance, or trust, are shown within the circular nodes). For example, if the assigned mission in Phase 2 turns out to be in the mountains, the estimate of trust generated by the incomplete tree is .55 (point *B* in Figure 8) , which is very close to the estimate of .53 in the more complete tree (point *B* in Figure 6). These two trees provide the same trust estimate at point *B* because Figure 6 is simply a refinement of Figure 8; the same experiences underlie both trees, but Figure 6 implies that the user has managed a finer discrimination among them. An important difference between the two trees lies in their ability to guide user behavior *after* point *B*. The user with the more complete tree in Figure 6 is likely to monitor and verify the aid's conclusions with respect to both angle of attack and rotorwash. The user with the incomplete tree in Figure 8 will not update his or her trust assessment at all.

Figure 9, on the other hand, is an incomplete event tree involving a hidden assumption, or potential *rebuttal*. If the assumption is false, the estimates of trust will be inaccurate; i.e., they will not be calibrated with respect to the true frequencies of correct performance. This user is not aware that angle of attack should be considered in judging the acceptability of the aid's conclusions. This could happen in a variety of ways. Perhaps the user's experiences were only in situations where angle of attack was not tactically important; perhaps the experiences happened to involve only cases where the recommended battle positions had favorable angles of attack; or perhaps the user's understanding of the aid's design is incomplete. We can think of these possibilities as reflecting a hidden assumption that angle of attack is irrelevant or, equivalently, that it is always ideal. In any case, the resulting trust estimates are excessively high at point *B*, compared to the estimates in Figure 6 and Figure 8. In addition, of course, the user will fail to verify angle of attack in evaluating the aid's recommendations, and will not update his or her trust accordingly. In effect, the user in Phase 1 has *already* inappropriately updated the trust assessment as if the ideal angle of attack had been verified.

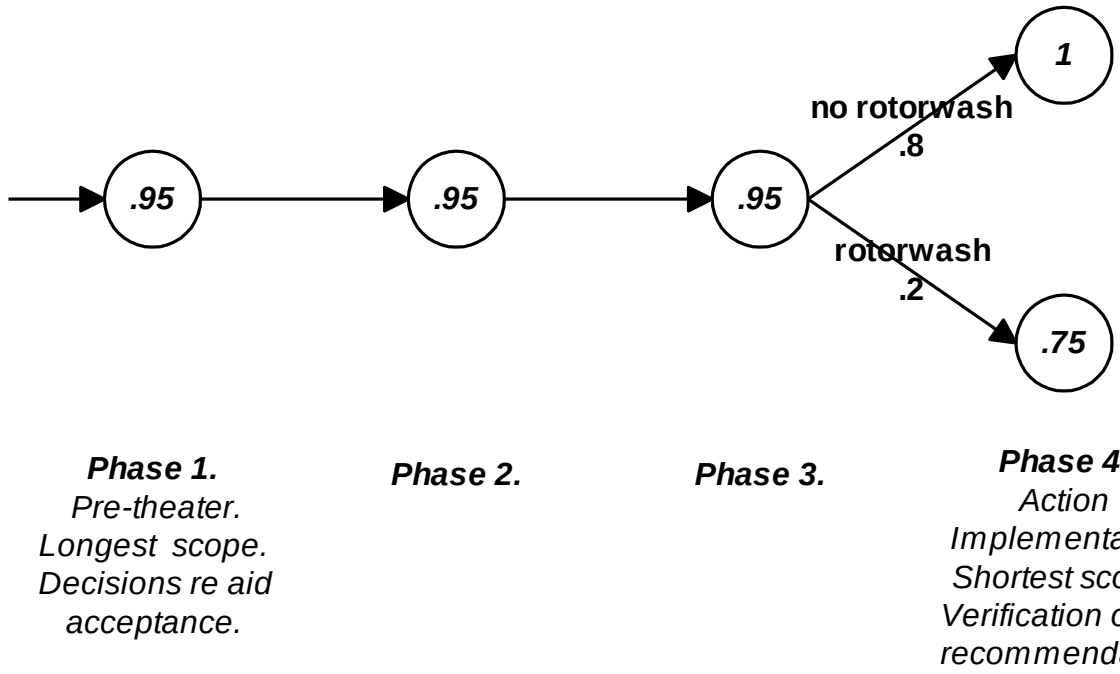

Figure 10. An even more incomplete, and inaccurate event tree. The user is not aware of the relevance of angle of attack, or the predictiveness of terrain for rotorwash. As a result, no updating takes place in Phases 2 or 3.

The event tree in Figure 10 represents an even more incomplete understanding of the decision aid. The only factor that this user knows about for assessing trust is rotorwash. As in the previous paragraph, the user's experience has

been confined to cases where angle of attack was irrelevant. This user's trust begins high in Phase 1, at 95%, like the user in Figure 9; however, it remains unchanged until verification of rotorwash in Phase 4. It is interesting to note that from the point of view of the user in Figure 9, who is also unaware of the significance of angle of attack, this user has an incomplete but *accurate* event tree. Both trees in fact are accurate given the implicit assumption that angle of attack will not be a factor (or will be ideal). Figure 9 allows more frequent updating of trust because it is a refinement of Figure 10, just as Figure 6 was a refinement of Figure 8.

Resolution. The resolution of a trust assessment tends to increase with completeness. In particular, the richer the event tree that faces a user at any given stage of aid use (i.e., the more nodes there are to the right of the user's current node), the higher the *expected*, or average, resolution for the remainder of the user's traversal of the tree. For example, the user at Phase 1 has an assessment of trust in the aid, but also anticipates collecting more relevant information by the time Phase 3 is reached. There is a set of possible trust assessments that this user could arrive at in Phase 3, depending on the outcomes of the information collection steps along the way. The more information the user plans to collect before getting to Phase 3, the higher is the average resolution of the possible Phase 3 assessments. In other words, *internal* Backing (deriving trust from branching possibilities to the right) yields higher expected resolution than *external* Backing (directly associating trust with a set of conditions).[7]

Table 3 shows, for users currently at Phase 1, the average resolution penalty they can expect at each successive phase of an event tree.[8] Four trees are compared: the "standard" tree, which we are assuming to be complete and accurate (Figure 6); an incomplete, but accurate tree (Figure 8); a tree that is both incomplete and inaccurate (Figure 9), and an even more incomplete and inaccurate tree (Figure 10).

In the complete and accurate tree (Figure 6), the chance of an acceptable recommendation — as viewed by the user before deploying to a particular theater — is fairly low (73%), and so is resolution (penalty = 20 out of a maximum of 25). Even after being assigned a mission and learning the nature of the terrain (Phase 2), there is very little improvement (penalty = 19). Learning about terrain has little effect because the two features that affect aid performance in this example happen to work against each other. Rotorwash is expected to be less a problem in the mountains than in the desert, while the aid is more likely to select a good angle of attack, by chance, in the desert than in the mountains. However, once mission implementation has begun and the aid has made an actual recommendation (Phase 3), the user can verify the angle of attack of the recommended battle position, drawing on expectations regarding the enemy's avenue of approach. This observation has a dramatic effect on trust regardless of its outcome, driving trust as high as 99% or as low as 32%. The expected resolution penalty at Phase 3 (penalty =

---

[7] It can be proven that average resolution is *always* improved by breaking down trust assessments into more specific branching possibilities — as long as the possible situations really are possible (have probability greater than 0) and as long as the original trust assessment was not already completely certain (probability not equal to 1.0 or 0). Here is part of the proof. At any given node in phase *n*, the user has an expected resolution for the next phase, *n+1*. Let G represent the actual relative frequency of the predicted event corresponding to the assessment value at any given node in phase n (e.g., if the user's assessment at this node is .75, let G be the actual frequency of the event across all the user's assessments of .75). Suppose G has only two daughter nodes. Let H and J represent the actual relative frequencies of the event corresponding to the assessments at the two daughter nodes of G, in phase n+1. Finally, let p_H and $p_J = 1 - p_H$ represent the probability of the observations leading to nodes H and J, respectively, given node G. Now the actual relative frequency at node G can be calculated from the actual relative frequency at the daughter nodes and their probabilities of occurring:

$$G = p_H H + p_J J .$$

The resolution penalty at G in phase *n* is, therefore:

$$R_G = (p_H H + p_J J) (1 - p_H H - p_J J) .$$

The expected resolution penalty for the daughters of G, given a user currently at G, is the probability-weighted average of the penalties that would be experienced at J and H:

$$E(R_{J,H}) = p_H H (1 – H) + p_J J (1 – J) .$$

It is easy to show that the resolution penalty does not increase from phase *n* to phase *n+1*:

$$R_G - E(R_{J,H}) = p_H p_J (H – J)^2 > 0, \text{ for } H \neq J \text{ and } p_H \neq 1, 0.$$

This effect holds regardless of whether the user's employment of probabilities is coherent, i.e., consistent with the axioms of the probability calculus.

[8] Resolution is closely related to the variance of the actual probabilities of the predicted event, across assessment groupings (see note 2). In fact, the resolution penalty as defined here is equal to that variance subtracted from the product of the overall probability of occurrence of the event times its complement. The latter term represents the uncertainty of the problem, independent of any assessments (Murphy, 1973).

11) is significantly lower than the penalty at Phases 1 or 2. Another bit of uncertainty, regarding rotorwash, is resolved when the user visually inspects the site (Phase 4). However, if either rotorwash or a frontal angle of attack is present, the resolution penalty will not drop to zero until the success the engagement with respect to battle position is actually known.

Table 3. Expected resolution penalty, assessed in Phase 1, for each future phase of decision aid use.

	Resolution Penalty (x 100) (max = 25; min = 0)			
Event tree	**Current Phase 1**	**Expected Phase 2**	**Expected Phase 3**	**Expected Phase 4**
Complete, accurate (Figure 6)	20	20	11	10
Incomplete, accurate (Figure 8)	20	19	19	19
Incomplete, inaccurate (Figure 9)	.20	19	19	19
More incomplete, inaccurate (Figure 10)	20	20	20	19

How is resolution expected to evolve in the accurate but incomplete tree (Figure 8)? In Phase 1 trust is approximately the same as in the complete tree, and so is the resolution penalty. New information about terrain is acquired in Phase 2, but this has little effect on trust or resolution, as we observed before. Since no new observations are made by this user after Phase 2, trust and resolution remain unchanged in Phases 3 and 4 — even though these were the occasions for the most dramatic changes in trust in the complete tree.

For the inaccurate tree (Figure 9), although trust is higher, resolution starts out in Phases 1 and 2 exactly the same as in the complete and accurate tree (Figure 6). The reason is that this tree makes precisely the same distinctions at these phases as the more complete tree. However, at Phase 3 this tree omits one of the crucial factors (angle of attack) for evaluating aid responses. As a result, in Phases 3 and 4, the resolution penalty does not decline in Figure 9 as it does in Figure 6. According to the knowledge represented in the two accurate trees (Figure 6 and Figure 8), an acceptable battle position must clear two thresholds (rotorwash and angle of attack) on features that the aid fails to consider, while in Figure 9 only one is recognized (rotorwash). For the same reason, the inaccurate tree provides a more dramatic reduction in trust as a result of learning about terrain in Phase 2, since terrain is correlated with a single factor, rotorwash, rather than with both. The user is unable to update trust at Phase 3, however, since information about angle of attack is not utilized. As a result, the user obtains a spurious sense of confidence or even certainty (trust = 100% or 75%) after verifying rotorwash in Phase 4.

Finally, the user in Figure 10 begins with the same high degree of certainty that the user in Figure 9 had, but no further observations are made to update trust until verification of rotorwash in Phase 4.

Calibration. Resolution is a measure of the *knowledge* a decision maker brings to bear on an assessment. Calibration is more a matter of the correct *labeling* of that knowledge. In other words, having distinguished a set of situations in which the probability of an event really does vary (resolution), has the decision maker correctly estimated the probabilities themselves (calibration)? Calibration is the correspondence of the estimated probabilities to real-world frequencies. Making the discrimination is far more important than correctly labeling it, and calibration penalty scores tend to be much smaller than resolution penalty scores.

Estimates of calibration require a standard of comparison, just as in the case of estimating completeness. How do we know what is the "true" probability of correct system performance at different points in an event tree? The answer once again is, of course, that we do not; it is worth repeating that no mental model of aid performance is ever likely to be complete or perfect. Once again, however, we can appeal to the most knowledgeable available sources: aid designers, instructors, long-time users of the aid, and potential users with experience in the relevant environments and missions. Their estimates can provide standards for comparison to a user's assessments of the likelihood of a successful system response. In these examples, we will again take Figure 6 as our illustrative standard.

Table 4 compares the same four event trees that were examined in Table 3. It shows the calibration penalty (squared deviation of assessments from the "true" probabilities) for the user of each of these trees prior to being assigned a mission (Phase 1). For the other three phases of decision aid use, it shows the expected, or average, calibration penalty, as viewed from Phase 1. Thus, the calibration penalty for users of Figure 6 are necessarily zero, since that figure is used as the benchmark for "true" probabilities.

These scores are quite low, even in cases where the user was not aware of important predictive features. Whether an assessment of trust is correctly calibrated or not always depends on the context in which the *user* intended it, i.e., on the user's decision tree and current location within it. Users with different knowledge regarding features that affect aid performance, hence, different event trees, will make different assessments of trust. But they may all be well calibrated within the *intended* contexts, i.e., given the features that they respectively recognize. For example, the user of the incomplete, but accurate tree in Figure 8 obtains only one new piece of information after Phase 1, viz., the nature of the terrain (Phase 2). The user fails to update trust assessments in Phases 3 (angle of attack) and 4 (rotorwash). Once this user finds himself or herself in desert terrain, trust becomes 75% and remains there. Nevertheless, this user is very nearly perfectly calibrated in every phase of decision aid use. The reason is, that the user's assessment of trust is understood relative only to the information actually relied on by the user, i.e., the system and the terrain. This user's assessments of trust in Phases 3 and 4 are identical to the assessment in Phase 2 (e.g., 75% in desert) because they continue to be conditioned on the same Grounds. Phase 3 and Phase 4 assessments pertain to the previous, more general context, rather than to the current one. The user ignores the newly available information about angle of attack and rotorwash, and continues to report (correctly!) the likelihood of a successful system response *across all desert situations*.

Table 4. Calibration penalty for users with different event trees. Scores are based on situations that are defined by features in the user's own event tree. But the "objective" probabilities for these situations are derived from Figure 6.

	Calibration Penalty (x 100) (max = 100; min = 0)			
Event tree	**Current Phase 1**	**Expected Phase 2**	**Expected Phase 3**	**Expected Phase 4**
Complete, accurate (Figure 6)	0	0	0	0
Incomplete, accurate (Figure 8)	0	0	0	0
Incomplete, inaccurate (Figure 9)	5	6	6	5
More incomplete, inaccurate (Figure 10)	5	5	5	5

This user's assessment, if given in Phase 3 after the aid has actually made a recommendation, might well be mistaken by others for an assessment of the likely appropriateness *of that recommendation* — including, for example, whether or not the recommended battle position implies a frontal attack. Similarly, if the user's trust assessment is given in Phase 4 after the recommended site has actually been examined, the assessment might well be mistaken for a judgment about the acceptability *of that specific site* — including its potential for rotorwash. As the user advances through these phases, the more specific assessments would in fact be far more useful. The large scope of the user's Phase 2 assessment becomes less and less relevant to the decision-making requirements of later phases. What users need is a probability of correct performance in the detailed circumstances that arise. To get this, users must increase the completeness of their event trees and make the observations required to advance along branches of the richer tree. Users' trust assessments will thereby be periodically recalibrated to match the frequencies of success within the actually existing situation. The resolution penalty is designed to capture this aspect of decision aid use. Notice that for this user, resolution penalty is twice as high in Phases 3 and 4 for the user of Figure 8 than for the user of the complete tree (Figure 6).

Figure 11 summarizes the relationships among APT parameters that we have discussed in this section. It shows how uncertainty about the quality of aid performance is reduced by increased resolution of assessments, increased calibration of assessments, and avoidance of unreliable assumptions. Resolution reflects the completeness of Grounds and Warrant, and is enhanced by training and experience that cuts across a variety of different systems, situations, and tasks. Completeness of the Grounds is also increased as users progress through phases of aid use from preparation to mission planning to mission execution, narrowing the temporal scope of their judgments as they advance. Calibration reflects the reliability of underlying knowledge, and is enhanced by in-depth experience and training with a specific system and within a specific context. Both kinds of experience, within and across conditions, can reduce the need for assumptions, and the susceptibility of trust assessments to unpleasant surprises, or rebuttals.

Comparison to Other Analyses of Trust

In this section, we briefly revisit previous theories of trust in order to describe their relationship to APT. Table 1 and Figure 2 demonstrate how APT accommodates the concepts described in previous models within a single coherent framework.

Grounds. A certain decision aiding system might perform well on routine, or *rule-based* tasks, but fail on more novel, or *knowledge-based*, tasks. Muir and Barber's notion of *technical competence,* which includes skill-based, rule-based, and knowledge-based levels, corresponds to one kind of condition that might affect the quality of system performance. In terms of predictions of correct system performance, it may be true, for example, for some systems that *p(correct action | task requires rule-based performance) > p(correct action | task requires knowledge-based performance)*. We have already seen, however, that this classification reflects only one of many variables that might condition system performance. For example, user expectations might be far more specific, such as predicting poor performance of a planning aid when tradeoffs among conflicting goals must be selected, or poor performance of a situation assessment aid when intelligence from higher headquarters is available, which the aid does not take into account. Expectations might also be more general, conditioned, for example, on a type of task; for example, predicting good performance of a medical expert system in handling infectious diseases but poor performance handling pulmonary disorders.

Predictions of performance can also depend on the system, system component, and/or system function that is involved. For example, when Muir and Moray (1996) introduced faults into an automated milk pasteurization plant, effects on user trust were differentiated by subcomponents of the plant (different pumps), but were not differentiated by function (display versus control). Expectations regarding system performance may include an entire class of systems, e.g., expecting good performance from automated sensors, but poor performance from expert systems: *p(correct action | system = automated sensor) > p(correct action | system = expert system*). Or they might be specific to particular aiding functions of a particular system. As we shall see in Chapter 0, relevant decision aiding functions include: data fusion, situation assessment, option generation, and option selection.

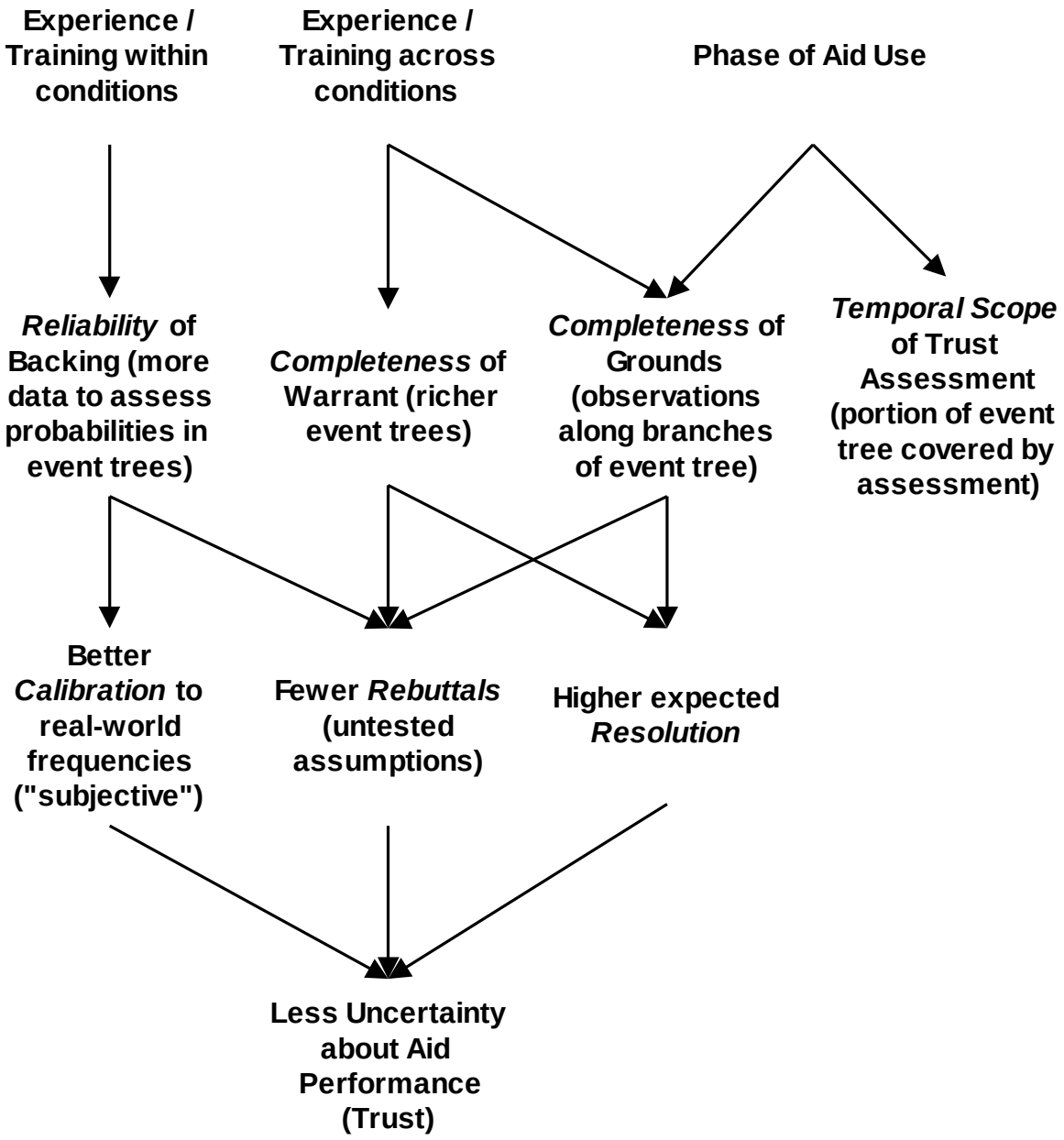

Figure 11. Relationships among APT parameters.

Backing. Knowledge required to predict aid performance may come from a variety of sources or strategies. Two important sources (identified by Zuboff) are *trial-and-error* experience with the aid and/or an *understanding* of the aid's design. Either of these, can provide good information about the probability of correct actions under various conditions. In combination, they can lead to flexible and accurate mental models of the aid. Muir's fiduciary responsibility seems to belong in this category, as a source of information about the aid's performance. It refers to a situation where little or nothing has as yet been learned (either from experience or knowledge of design) about the conditions of good performance for a particular system. One recourse, Muir suggests, is to fall back on an assumption of the good motives of the designers (*fiduciary responsibility*). Note that this strategy must also assume the technical competence of the designers. Also note that there may be other fall-back strategies in the case of a new or poorly understood aid. Another possible fall-back is a worst-case assumption that the aid will not operate effectively. Another is reliance on the reported experiences of other aid users, its reputation, so to speak. Still another source of information, for better or worse, is a projection of animate properties onto the aid, such as human-

like beliefs and desires. Although they vary in the quality of the information provided, each of these constitute a type of Backing: sources, methods, or models that users can employ to generate predictions of aid performance.

Completeness / Resolution. Muir represents the evolution of trust with experience as a progression from *predictability* to *dependability* to *faith*. This progression seems to presuppose only good news about the performance of the aid. If the user is fortunate, the aid will perform well under most or even all conditions, but this is not necessarily so. More generally, a user may learn conditions of both good and bad performance. The metrics we propose are independent of the results. Nevertheless, we can interpret Muir's ideas in terms of *completeness*. Predictability refers to the case where the performance of the system has been observed (or inferred from design knowledge) only in a limited range of relevant conditions — but the performance (probability of a correct action) of the aid has been high in all these conditions. Dependability refers to the case where the performance of the system has been observed or inferred to be good under a wider range of potentially degrading conditions. Faith seems to involve a qualitative leap rather than additional increases in completeness. It might involve a higher-order inference across conditions: The aid has been observed (or inferred) to perform well in so many different kinds of conditions, that the user predicts good performance *everywhere*.

Temporal Scope. Muir's category of *persistence* refers to individual differences in basic expectations about the consistency of nature, human life, and society. Persistence can be included logically in our framework as a still higher level prediction of performance, with temporal scope generalized to include all physical, biological, or moral "systems" for "all time." Such a broad scope would correspond to a "phase of decision aid use" before one had actual contact with any decision aid (Figure 7). As far as we know, no one has as yet demonstrated that differences in beliefs about persistence in this sense affect trust in automation, and our research does not address this level.

In sum, the framework we have described has a number of significant advantages:

1. It accounts more clearly than existing theories for important distinctions regarding trust. For example, it clarifies the distinction between faith and fiduciary responsibility, which Lee and Moray regarded as the same. Both involve the adoption of assumptions in assessing the performance of the system. But faith extrapolates from relatively complete knowledge about system performance, while fiduciary responsibility involves a state of relative ignorance. Similarly, this framework distinguishes between predictability, which is a level of completeness of knowledge about good system performance, and competence, which is a feature of task demands that is correlated with system performance. It similarly explains other distinctions that Lee and Moray questioned (see Table 1).

According to Muir, trust is decomposable into two dimensions: Types of expectation, and basis of expectation. However, our analysis suggests that the three levels identified by Muir and Barber for the types-of-expectation "dimension" are better construed as pertaining to three different dimensions. Persistence pertains to the *temporal scope* of a trust assessment, competence to the *Grounds* and *Warrant* for a trust assessment, and fiduciary responsibility to the *Backing*, or source of knowledge, for the assessment (viz., assumptions about the good intentions of designers). On the other hand, the three levels identified by Muir and Rempel for the basis-of-expectation "dimension" (predictability, dependability, and faith) do correspond to a single dimension, namely, degrees of *completeness* of the grounds and warrant for a trust assessment. Finally, the three aspects of trust in Zuboff's model (trial and error, understanding, leap of faith) also correspond to a single dimension: sources of knowledge, or Backing, for the assessment. All nine concepts in Table 1 are related to a single relational structure in Figure 2.

2. At the same time, the APT framework is more complete than previous models of trust. For example, none of Muir's (and Barber's) "types of expectation" provides a full account of the concept to which it refers. Persistence is only one level of temporal scope (the longest), referring to a generalized belief in regularity in nature and society. Other levels of temporal scope within which a trust assessment may be framed include a particular output of the aid (the shortest), the aid's functioning in a particular situation, and the aid's functioning across an entire domain. The latter in fact are probably more pertinent to understanding real-world assessments of trust than persistence. Similarly, competence — which Muir and Barber break down into skill-based, rule-based, and knowledge-based — is far from exhausting the set of features of a situation or a system that may be correlated with the quality of its performance. Finally, fiduciary responsibility (which is a specific type of assumption) is far from exhausting the sources of belief upon which an assessment may be based. In addition, of course, APT provides for other concepts, such as reliability, resolution, calibration, and rebuttals that are not addressed in the other frameworks.

3. APT's measure of trust itself is likely to be more useful as a basis for training decision strategies, as we shall see in the next section and in the next chapter. One reason is that APT uses a well-understood measure of trust (probability) rather than an ad hoc one. This measure, the probability of correct system response, can be instantly plugged into decision theoretic models that provide insight into how users should interact with an aid. (In such benchmark prescriptive models, the optimal action is determined by maximizing expected utility over uncertain

events.) For example, the lower the resolution of a trust judgment, the more desirable it is that the user should take time to verify an aid's recommendations before accepting *or rejecting* them.

4. The final advantage of APT is increased clarity in the task of assessing and training trust. APT resolves the numerous ambiguities that have plagued attempts to elicit subjective assessments of the different dimensions of trust (e.g., asking subjects to assess "competence," "reliability," "faith," "dependability," "predictability," and so on). It seems clear that users are not able to unambiguously interpret these terms, as evidenced by Muir's finding that faith, dependability, and predictability (as subjects interpreted them) developed in an order opposite to that predicted. APT, on the other hand, stipulates that users provide all assessments in a single clear, consistent format: *the probability of a correct conclusion by the aid*. We can vary the temporal scope of the required assessment (e.g., the probability that the response in this problem will be correct versus the proportion of correct responses over a period of time in a given situation, versus the proportion of correct responses over all situations and aid functions). We can also vary specific features of the situation and the aid, and in this way determine the features (or the details of the mental models) that in fact affect subjective trust at different scopes. At the same time, we can provide convergent validity by asking users what factors in the aid and the situation influenced their judgment of trust. The basic underlying concept, however, remains consistent and clear through all these variations. As we shall see in the next section, on training implications, well-established methods for training subjective judgments of probability can be employed (e.g., Goodman, 1972; Stillwell, Seaver, & Schwartz, 1981; Von Holstein, 1975).

Training Implications

The most important implications of the APT framework are for training development. Indeed, various aspects of the model lend support to different elements of a comprehensive training strategy for decision aid users. For convenience, we will adopt the terminology developed by Salas & Cannon-Bowers (1977) to describe this training. According to them, a training *strategy* orchestrates *methods* (such as instruction and practice) and *tools* (such as simulation and feedback) to convey a *content*. Table 5 summarizes the implications of this chapter (and the next) for such an overall training strategy.

First, APT helps identify training requirements, or objectives regarding the content that is to be conveyed. Based on the present chapter in particular, we can identify four contributions of APT to the definition of training requirements and the specification of training content:

- Mental models of decision aid knowledge
- Critical thinking about decision aid performance
- Decision-aid driven situation awareness
- Probabilistic assessment of trust in decision aids

Figure 12 shows how each of these contents is related to elements of the APT framework. In addition to content requirements, APT provides *tools* for fleshing out the specific content of training in a domain and for measuring the success of the training. These elements can be roughly classified according to whether they flow from the qualitative or the quantitative aspects of the framework.

Qualitative Training Content: Mental Models and Critical Thinking

Mental models to represent decision aid knowledge. Causally and temporally organized event trees provide a rich framework for capturing the kind of knowledge required in effective decision aid use. These structures spell out the features that are predictive of successful decision aid responses and that are required for user decisions about reliance on the aid at each phase of use. They indicate when such information becomes available, and provide a clear mechanism for relating the information to assessments of trust. In particular, information available at Phase 2 may be used for decisions about automation mode, e.g., whether or not to monitor an aid's recommendations carefully. Assuming that the user does chose to monitor the aid, information available at Phase 3 may then be used to make decisions about acceptance, rejection, or verification of specific aid recommendations.

Table 5. Outline of a training strategy for decision aid users based on APT.

Strategy	Tools	Methods	Content
Trust-based Decision Aid User Training	Cognitive task analysis (critical incident interviews to flesh out event tree and interaction strategy models) Simulation (scenarios based on event tree and interaction strategy models) Feedback (based on event tree and interaction strategy models) Performance measures (based on APT parameters and interaction strategy models)	Information-based: Lecture and discussion Practice-based: Guided practice, behavior modeling, scenario-based simulation	Mental models of events affecting decision aid performance Critical thinking strategies for novel situations Dynamic situation awareness of mental model elements & signs of trouble Probabilistic assessment of trust based on mental model and situation awareness Strategies for interacting with decision aids based on trust

Critical thinking about decision aids. In addition to the substantive knowledge about the aid represented in event trees, skilled users will also be adept in handling novel or unanticipated situations. Novel conditions may occur that were specifically anticipated neither by aid designers nor by user training. Critical thinking skills can help users learn to "expect the unexpected" and to handle it effectively when it occurs. For example, the following two critical thinking strategies may supplement the knowledge embedded in mental models:

1. *Detecting and handling conflict.* Conflict among different sources of information or arguments about trust can be a symptom of erroneous assumptions in a user's mental model of the decision aid, or in the user's understanding of the situation. For example, observations of actual aid performance under various conditions may violate the expectations generated by an event tree, or there may be a surprising difference between an aid's conclusion and the user's independent judgment. Situation awareness must be expanded to include such symptoms of trouble, as we shall see below. Users can be trained to be alert to such conflicts and to use them as opportunities to learn more about the situation and the system (Cohen, Freeman, & Thompson, 1997).
2. *Devil's advocate.* When stakes are high and time is available, devil's advocate strategies can be effectively employed for uncovering hidden assumptions in mental models and generating alternative interpretations of events. In such a strategy, users try to generate arguments against a favored conclusion. For example, users may imagine that a conclusion of their own or of the aid is false, and to explain how that could be so. Such strategies have been found to be an effective countermeasure against overconfidence (Koriat, Lichtenstein, & Fischhoff, 1980) and to be successfully trainable in realistic operational settings (Cohen, Freeman, & Thompson, 1997).

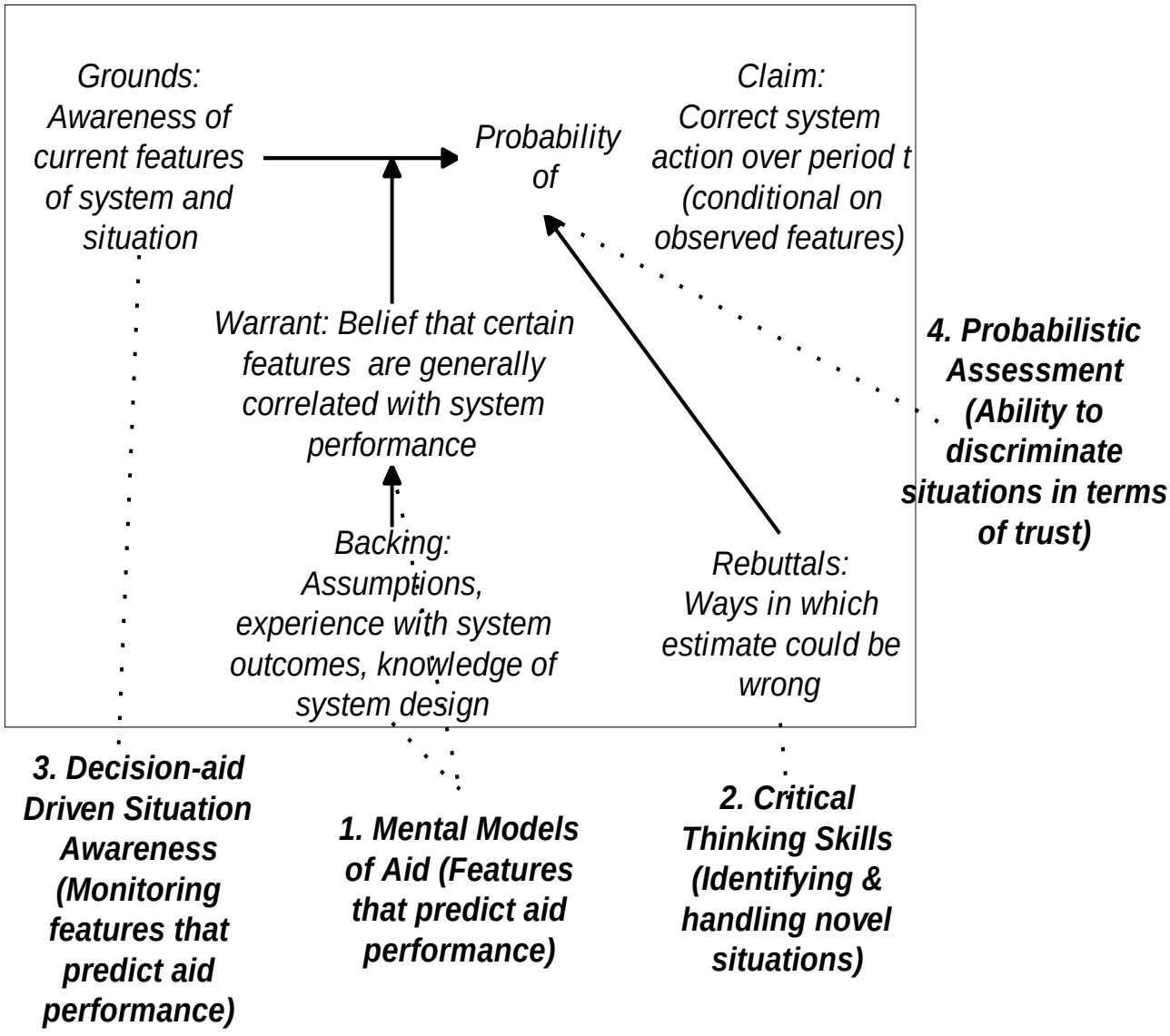

Figure 12. Training requirements generated by APT, and their relationships to components of the APT framework.

Dynamic decision-aid driven situation awareness. A corollary to the development of adequate mental models of aid performance is the development of situation awareness required for applying those models. Another training requirement, therefore, is to help users develop the monitoring and observational skills necessary to track events that are diagnostic of decision aid reliability. In other words, situation awareness for proficient decision aid users must be at least partially driven by the requirements of effectively interacting with the aid.

Such situation awareness is relevant at every phase of decision aid use. For example, at Phase 2, users may need to monitor for information that could suggest the need for a change in automation mode. For example, if conditions occur that are correlated with relatively poor aid performance, users may need to verify the aid's conclusions more closely or even switch to manual mode. Similarly, at Phase 3, users need to monitor for information signaling that more thorough verification of an aid conclusion is appropriate.

Event trees help define the features that should be monitored for at each of these phases. In addition, however, users may need to monitor for situations that call for critical thinking, for example, when the degree of novelty or uncertainty in the situation suddenly increases. By definition, such situations cannot be anticipated ahead of time in an event tree.

Recent research has emphasized the heavy workload demands that monitoring automated systems imposes (Parasuraman, 1996). In some cases, monitoring for system problems is equivalent, or nearly equivalent, in effort to manual performance (the "automation paradox"). An important training requirement for decision aid users involves

learning effective and efficient monitoring strategies. The following is a more concrete example of the form such a training requirement might take. The example will illustrate the importance of identifying key features that predict weaknesses in system performance. These features may be conditions anticipated in an event tree (i.e., the *Grounds* and *Warrant* of trust assessments), or they may be signs of trouble (such as conflicting evidence or goals) that trigger critical thinking. With knowledge of such features, users do not have to painstakingly verify every conclusion the system reaches, but can concentrate on the conclusions that are most likely to be problematic.

For some aid functions and situations, it may be sufficient to monitor for very specific indicators of problems, or cues to critical thinking. As an example, we take a data fusion device that correlates and aggregates information from different sensors. An example of such a device is a system for combining estimates of target range obtained from different sensors and solution algorithms. A plausible method for aggregating different estimates is to use Bayesian updating. Given certain simplifying assumptions, the result is a weighted average of the individual estimates, with weights estimated from the inverse of the covariance matrix (Cohen & Brown, 1980). The fused data can be represented by a single estimate with an uncertainty interval that is smaller than the uncertainty of any of the individual estimates. In other words, the pooled solution should be more accurate than any of the individual solutions. This method (or similar but more sophisticated variants) for the most part produces satisfactory answers. In at least one case, however, it is not trustworthy. When two estimates are each regarded as trustworthy by the aid (i.e., they have small uncertainty intervals) but are widely separated, the aggregated estimate is a value falling between the two with an even smaller uncertainty interval. Yet it seems implausible, when facing two widely discrepant estimates of target range, to accept a value between the two estimates with a high degree of confidence, which neither of the techniques regards as at all likely! In this case, the method of statistical aggregation has been misapplied: The assumptions underlying one or more of the discrepant estimates may be erroneous. In another words, instead of simply aggregating the estimates, one should try to determine which estimate(s) are invalid in this situation, and drop them.

This example makes clear that an operator need not rework the entire aggregation problem every time he wishes to verify the performance of a fusion aid. Instead, he might look for *conflict*. In particular, he can quickly check the distance between the most discrepant pair of estimates (in comparison to their uncertainty intervals). If the distance is small, the aid's solution is trustworthy; if large, and if the costs of an error are high, he might revert to manual mode and try to discover the cause of the discrepancy in order to adjust the automated solution.

Similar examples can be presented for situation assessment (where conflicting cues regarding enemy intent, for example, might be aggregated by an aid) and for planning (where conflicting goals might be satisfied by recommending an unsatisfactory compromise option). As an example from planning, an option selection aid (such as the Combat Battle Position Recommendation aid in RPA) may recommend a particular plan (e.g., an engagement area and set of battle positions) based on a variety of algorithms that aggregate multiple goals or evaluative dimensions. For example, the aid might evaluate alternative engagement areas in terms of factors like obstacles to enemy movement, opportunity for long-range fires, continuous target visibility, and the availability of good attack routes from holding areas (i.e., routes with good cover and concealment and prominent terrain features for navigation). Recommendations are highly trustworthy as long as all or most of the evaluative criteria agree. Suppose, however, that candidate engagement area A is far superior to area B in terms of obstacles to enemy movement and continuous visibility, but all attack routes from the holding area to A would expose the attack helicopter company to possible detection. Engagement area B, on the other hand, is less favorable in terms of battle positions, but offers ingress and egress routes with better cover and concealment. Suppose the decision aid weights features of the engagement area more heavily than features of the attack route and thus recommends engagement area A.

The commander may not be satisfied simply to accept the alternative that the aid recommends. For example, if he is planning to employ massed fires in a maximum destruction attack, surprise may be more important to him than the ability to sustain a prolonged attack. He may thus prefer engagement area B with its superior attack routes. In a case where goals conflict, he will probably want to know what goals are being sacrificed, and what alternatives are available that represent different tradeoffs. Here, as in the earlier example, the user need not reexamine all the reasoning underlying every aid recommendation. He need only be on the alert for goals that *conflict* with the recommended option. When they exist, he may choose to explore alternative candidates in manual mode.

In Chapter 0, we advocate measures of situation awareness that include not only knowledge of the current and predicted values of variables in the situation, but also *knowledge of what variables are and will be critical*. Situation awareness does not require knowing everything, but does require paying attention to what is important. Decision-aid driven situation awareness illustrates this point.

User-decision aid interaction strategies. An important content of decision aid user training involves reliance decisions — i.e., decisions regarding how much responsibility to *entrust* to the aid. Such decisions arise at each phase of decision aid use, and will be the topic of the next chapter.

Qualitative Training Tools: Interviews, Scenarios, and Feedback

Elicitation of mental models to build training content. As noted, a crucial goal of training is to transfer effective mental models to decision aid users. The content of training will be determined by elicitation of event trees from experienced decision makers in the domain, from aid designers, and from experienced users of the aid (although the latter may not be available in the case of a newly developed aid). A variety of elicitation methods can be used for this purpose, and information from these different groups may be combined in the design of decision aid user training:

1. Interviews with unaided, experienced decision makers might utilize the *critical incident* method, in which they are asked to describe challenging experiences in depth, either in combat or exercises. In addition, simulated problem scenarios (such as map exercises for locating attack helicopter battle positions) can be presented and decision makers asked to think aloud as they work. In both of these techniques, elicitors work with interviewees to identify decision points, key information, and the decision making strategies utilized for each decision. More general follow-up questions can elicit a sense of the importance and relative likelihood of occurrence of each type of decision.
2. Interviews and observations of decision makers using the decision aid can shed light on which aspects of performance seem successful and which do not, and can shed light on the mental models of the aid that current users employ.
3. In parallel, training designers will also work with decision aid designers, and examine relevant documentation, to learn about the key decisions that were intended to be supported by the aid, and the information and processing methods and assumptions utilized by the aid.
4. Aid designers will combine the results of interviews and observations of unaided and aided decision makers and with aid designers to generate predictions regarding strengths and weaknesses of the decision aid relative to its human users, and how often these strengths and weaknesses will be manifest under different conditions. For example, strengths might include an ability to quickly synthesize a large amount of data. Weaknesses might include omission of inputs which are sometimes important, or assumptions embedded within the processing algorithm that on some occasions may turn out to be false.
5. Decision aid strengths and weaknesses, as well as advance indicators of such strengths and weaknesses, will be classified in terms of the temporal phase at which they are likely to become manifest. Relevant features of the system, mission, task, or aid conclusion will be identified and organized into event trees. Interviews with decision makers or with decision aid designers will be used to estimate probabilities of these various features' occurring.
6. Event tree models can be confirmed or disconfirmed in interviews with experienced decision aid users, if they are available. Interviews might use the critical incident method to probe for situations in which the aid's performance was exceptionally good or bad. In addition, experienced users might be observed solving problems and interacting with the aid in scenario-based simulations. These observations can also be used to identify areas where the performance of current aid users (if any) does not reflect appropriate trust, e.g., where users show over-reliance or under-reliance on the aid.
7. A final test of the models is whether instruction succeeds in improving the performance of less experienced decision aid users, and whether the points of improvement correspond to the key nodes in the relevant event tree models.

Scenarios and feedback. Event tree representations are useful in the construction of training scenarios and the design of feedback. The sequence of significant observations regarding aid performance that is represented in the event tree can serve as the basis for the design of scenarios that vary the features of the system, mission, task, and/or aid conclusion, and observe the effects on the assessments of trust and the decision aid reliance decisions that participants make (such as selection of automation mode or acceptance/rejection of an aid conclusion). Debriefings can use event tree representations to provide specific feedback regarding trainee's performance with respect to features of the event tree that they failed to respond to or responded to inappropriately. (We will discuss feedback regarding decision aid interaction strategies in the next chapter.)

Quantitative Training Content: Probabilistic Trust Assessment

One feature of APT is the use of probabilities to measure trust. But it is important not to misunderstand or overemphasize this quantitative feature of the model. The advantage of probabilistic concepts in this context is not for rigid modeling, or formulaic adherence to "normative axioms." Nor do we believe that real-life decision makers can or should be induced to convert their natural thought processes into probabilistic forms (see, e.g., Cohen &

Freeman, 1996). The real benefit of the probabilistic formulation, in fact, is in its contribution to clarity of thinking and to the development of more naturalistic training materials. For example:

1. Probabilities clarify the pragmatic essence of the concept of trust, which is a prediction about decision aid performance. They highlight and sharpen understanding of the uncertainty of such predictions. And they dramatize how predictions evolve as a user advances along a path in an event tree.
2. As a corollary of the above point, probabilistic notions emphasize the concept of decision aid fallibility. Users learn to understand that no matter how well-designed a decision aid is, its recommendations or conclusions may sometimes be wrong. The probabilistic character of trust focuses attention on uncertainty about aid accuracy in specific situations. And it prepares the ground for learning about strategies for allocating tasks between users and decision aids based on expected accuracy (as discussed in the next chapter).
3. Probabilities motivate users to discriminate among situations that differ in their implications for aid performance. Probabilities, from this point of view, may be thought of not as numbers but as labels for different types of situations. There is behavioral evidence that discriminations made with numerical scales tend to be more refined than discriminations made with verbal labels.
4. Probabilities are a convenient way of scaling the importance of different factors in a situation. In an earlier example (Figure 6), knowledge of terrain was not as important as expected because of competing interactions with rotorwash and angle of attack; by contrast, there was a much larger impact of angle of attack on predictions of aid performance. These orderings of attentional priority are almost certainly more important than the literal values of the probabilities, which in most cases cannot be known precisely anyway. The importance of the prioritization function is reflected in the resolution penalty score (Table 3), which focuses on probability purely as a device for discriminating situations, rather than the far weaker calibration score (Table 4), which measures the closeness of assessments to the "true" values.
5. Probabilities of successful performance lend themselves readily to incorporation in strategies for allocating tasks between users and aids (see next chapter).
6. Finally, probabilities provide a unique training tool for shaping trainees' intuitions and behavior in the direction of more expert performance. This is the topic of our next discussion, on diagnostic measures.

Notice that the first three of these functions are essentially qualitative. Moreover, the most significant benefits of the last three functions come from their use in generating scenarios and feedback for teaching users qualitative, pattern-recognition skills.

Quantitative Training Tools: Diagnostic Measures

Diagnostic measures for training. The objectives of training decision aid users include:

1. the ability to generate and use an event tree that represents the knowledge of a proficient user regarding factors that affect aid performance, and
2. the ability to discriminate situations arising in the event tree in which successful performance is more or less likely.

Probabilistic event tree parameters can provide useful quantitative measures of the effectiveness of training in conveying this knowledge to trainees. Among the measures of training impact that might be used are the completeness of trainees' mental models relative to the standard event tree, and the resolution and calibration of trust assessments.

A particularly interesting type of diagnostic measure, which combines both resolution and calibration of trust assessments, is a *proper scoring rule*. The most familiar proper scoring rule is the Brier score (1950). On each trial, the trainee produces a probabilistic assessment (for example, a prediction of the correctness of the aid's response), and feedback is provided based on the actual outcome. Across trials, the trainee tries to minimize a penalty score equal to the square of the difference between the probability assessed by the trainee and 1 if the event occurred, or 0 if the event did not occur. For example, if the trainee assesses trust at 60% and the aid's conclusion is in fact correct, the penalty would be $(1 - .60)^2 = .16$. If the aid's conclusion had turned out incorrect, the penalty would have been $(0 - .60)^2 = .36$. The defining feature of a proper scoring rule is that the trainee can minimize the overall penalty (i.e., maximize reward) by reported her or her true best judgment regarding the probability. A linear scoring rule, for example, is not proper, because the optimal strategy is to report a probability of 1.0 if you believe that the true probability is greater than .5, and to report 0 if you believe that the true probability is less than .5.

It turns out that the quadratic feedback function can be decomposed into additive components corresponding to the measures of calibration and resolution that we discussed in this chapter (Murphy, 1950). Table 6 shows the Brier scores that would be earned by trainees using each of the different event tree that we discussed. These values

represent the sum of the scores in Table 3 and Table 4. They give a clear advantage to the complete and accurate event tree of Figure 6, and a small advantage to the incomplete but accurate tree of Figure 8.

Assessments of trust in an aid can be elicited at each phase of a scenario, and across different scenarios that vary in relevant features of the mission, task, situation, and aid conclusion. Trainees can be given feedback that applies proper scoring rules to the user's assessment in the light of the actual outcome of the aid's recommendation. Such procedures have been highly successful in improving judgments of uncertain events (Lichtenstein, et al., 1982).

Table 6. Results of feedback with a proper scoring rule for different event trees.

	Brier Score (x 100) (max = 125; min = 0)			
Event tree	**Current Phase 1**	**Expected Phase 2**	**Expected Phase 3**	**Expected Phase 4**
Complete, accurate (Figure 6)	20	19	11	10
Incomplete, accurate (Figure 8)	20	19	19	19
Incomplete, inaccurate (Figure 9)	24	25	25	26
More incomplete, inaccurate (Figure 10)	24	24	24	24

TRUST AND USER-DECISION AID INTERACTION

In the previous chapter we described trust as an evolving assessment of the performance of a decision aid. Trust changes as users advance through an event tree, acquiring more experience with the aid and observing conditions that are known to predict its performance. However, trust in an aid may evolve over time not only because of the passive flow of events, but because of active decisions by the user to seek new information and to use that information to improve performance. As users gain understanding of the aid's strengths and weaknesses, they also learn how to interact more effectively with the aid, compensating for aid weaknesses and exploiting aid strengths to reduce their own workload and improve performance. As a result of their own active participation, users' trust in an acceptable outcome is likely to increase. Trust must now be considered a product of the *interaction* between the decision aid and the user. In this chapter, we extend the model of trust presented in the last chapter to account for user-decision aid interaction, and we explore the implications for training.

A principal advantage of the Argument-based Probabilistic Trust (APT) framework, as we have noted, is that it lends itself readily to incorporation into systematic, benchmark models of user interactions with decision aids. Such models provide training content, helping to define training requirements for user-decision aid interaction strategies. And they also provide training tools, since they can be used for the development of training scenarios in which interaction strategies are elicited, practiced, and provided feedback. Table 5 in the previous chapter summarizes these roles.

Figure 13 shows the relationships among variables that might be expected to influence a user's reliance on an aid, including trust. Many elements and relationships in the figure are from Riley (1989), and are based on empirical studies of human interaction with automated systems. The elements have been rearranged (without disturbing their relationships) so that the horizontal dimension corresponds roughly to time or causality, and the vertical dimension to the user versus the decision aid. The figure thus brings out temporal flow (roughly from left to right), and the approximate symmetry between the variables pertaining to user performance (at the top) and those pertaining to aid performance (at the bottom).

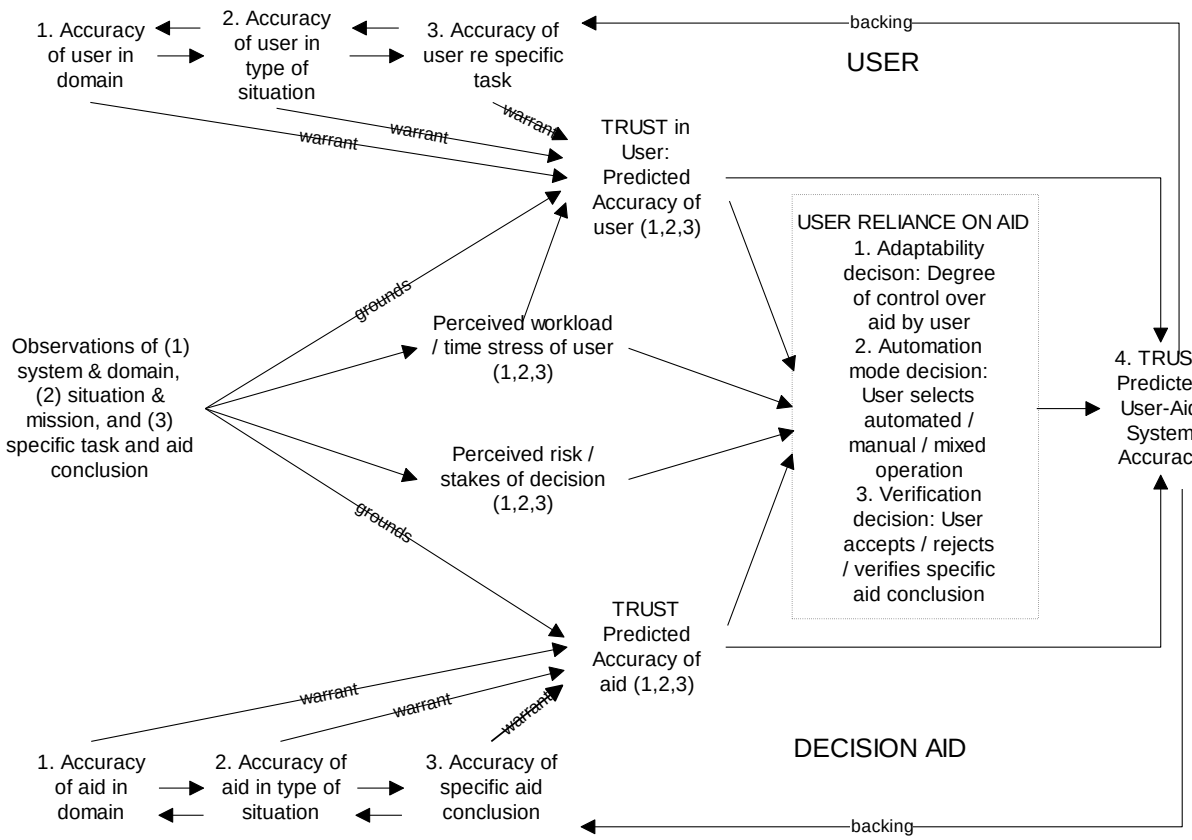

Figure 13. APT-R, i.e., Argument-based Probabilistic Trust in the context of user Reliance decisions. Numbers represent decisions at different temporal phases, and the factors that affect them in the corresponding phase.

Figure 13 addresses decisions about user-decision-aid reliance in three broadly defined phases, corresponding to the cycles of different temporal scope that we discussed in the last chapter. Each phase is associated with different decisions about the mix of user and decision aid performance:

1. Pre-theater: Many decisions made in this phase are by managers, designers, and trainers , although they can (and should) draw input from users. These include: (i) whether and how an aid should be built and deployed for a particular domain, (ii) the types of interaction and automation modes that it should permit, including the degree to which the aid adapts automatically to the user and the degree to which the user can adapt the aid; (iii) how training should be focused across scenarios and aid applications, e.g., where the most difficulty is anticipated. In addition, users who receive training on the aid make judgments regarding overall aid acceptance, and may develop expectations regarding its performance in different types of situations.
2. Mission planning: In Phase 2, the user makes choices that influence the aid's performance over an extended period of time, e.g., over the course of a mission, by (i) selecting the automation mode that will be employed, (e.g., fully automated, monitoring and verification of the aid by the user, manual performance with verification of the user by the aid, and fully manual), and (ii) adjusting aid parameters, modifying rules, or constraining aid solutions in anticipation of a particular context of aid use. The opportunity for users to make these Phase 2 decisions is contingent on Phase 1 design decisions that include the user's ability to adapt the aid.
3. Mission Execution: In Phase 3 users make decisions about how to handle a specific aid conclusion or recommendation, (e.g., whether to accept it, override it, modify it, or take time to verify it). The opportunity to make this Phase 3 decision is contingent on a Phase 3 automation mode decision to monitor aid recommendations.

These phases correspond to three levels of the temporal scope parameter in the APT model of trust (Figure 2), and to the phases in decision aid use depicted in the various event trees in the previous chapter. The boundaries between

phases are sometimes fuzzy, for example, when a user changes automation mode frequently during the execution of a mission. This fuzziness is not particularly important, since the same basic principles of user-aid interaction will be found to apply for each kind of reliance decision, regardless of when they occur. Nevertheless, the division of such decisions into different phases highlights certain important differences. Each phase draws on knowledge relevant to that phase (e.g., regarding the entire domain of the aid, a specific mission and situation, or a particular task and aid conclusion), is influenced by estimates of trust, stakes, and workload over a different temporal period, and has effects that usually last for different periods of time. Choices made in the longer decision cycles are revisited less frequently, and determine the options that are available to users at the shorter cycles.

In Phase 4, the outcomes of system and user performance become known. As shown in Figure 13, Phase 4 performance provides feedback that can influence judgments of trust in *future* cycles of (1) decision aid design or training, (2) planning for new situations or missions, and (3) acceptance or rejection of subsequent aid recommendations.

In the previous chapter, on the APT model, we focused on a subset of Figure 13: the node labeled *trust: predicted accuracy of aid*, and the nodes that influence it (as shown by the four arrows pointing toward it). These elements in Figure 13 map directly onto parts of the APT model in Figure 2. *Trust: predicted accuracy of aid* represents the *Qualified Claim*; observation of features of the system, domain, situation, mission, task, and/or conclusion serve as *Grounds* for that claim; the user's knowledge regarding the accuracy of the aid for the relevant domain, situation, mission, task, and conclusion serves as a *Warrant* that links Grounds and Claim; and the user's experience of the aid's past performance, represented by a feedback loop from Phase 4 performance, constitutes one kind of *Backing* for the Warrant. (Other kinds of Backing, such as design knowledge or assumptions, are not shown in Figure 13.) The top of Figure 13 shows how these same components apply to trust by users in their own performance.

In this chapter and in Appendix A, we expand our focus to include all of Figure 13, starting with the broader concept of trust represented by the node labeled *trust: predicted user-aid system accuracy*. Working backwards (from right to left in Figure 13) trust in the overall user-system interaction is a function of the user's self-trust, the user's trust in the aid by itself, and the way performance by the user and aid are mixed in the decision making processes. This broader conception of trust, then, includes both trust in oneself and trust in the aid, and the *reliance* decisions that select between, or blend, them. Since Figure 13 shows how the Argument-based Probabilistic Model of Trust can be extended to reliance decisions about user-aid interaction, we will call it the APT-R framework.

Continuing to work backward, decisions about user-aid interaction at each phase – i.e., *user reliance on the aid* – are influenced by (i) trust in the aid, (ii) user self-trust, (iii) the expected workload or time stress on the user, and (iv) the expected stakes, or cost of an error. For example, Lee and Moray (1994) found that the user's choice to override or accept a particular automated solution (i.e., a reliance decision at Phase 3) is a function of a comparison between the user's confidence in his own performance and trust in the performance of the machine. As Figure 13 indicates, decisions about automation mode in Phase 2 are also a function of self-confidence and trust in the aid, but predicted over a longer time interval, and decisions about decision aid adaptability and training in Phase 1 are based on the same factors (trust in the aiding technology versus trust in the users), but estimated over the lifetime of the aid. We shall pursue these ideas further in this chapter in our discussion of benchmark models. We will begin by laying out a framework for analyzing the verification decision in Phase 3, and then work backwards to the automation mode decision in Phase 2.

Like the APT model of trust in the aid in the previous chapter, the APT-R framework has both a qualitative and a quantitative aspect. The qualitative aspect involves an expansion of the representational format of event trees, to include user decisions. The quantitative aspect draws on the decision theoretic concept of value of information. Once again, the main value of the quantitative aspect is to train pattern recognition, rather than analytical processes: to induce finer discriminations by users among factors that should affect their interaction with the aid.

A Model of the Verification Decision

The user must have already followed a particular path in an event tree in order to have an opportunity to make a verification decision. As noted, a user can chose to monitor an aid in Phase 2 only if aid designers in Phase 1 chose an adaptable design. In addition, the user will still not face the verification decision in Phase 3 if the user did not choose in Phase 2 to monitor the aid. Finally, the verification decision is only possible after the aid has made a specific recommendation, or drawn an specific conclusion, in a specific task. The decision aid user either chooses to make a final decision regarding this aid recommendation (by accepting it, modifying it, or rejecting it), or else the user chooses to consider it further. This is the verification decision.

Each reliance decision (e.g., to monitor the aid, or to verify a specific conclusion) is based on all the information the user has collected along the path in the event tree up to that point. The verification decision, therefore, is based on information about the system, domain, and situation obtained prior to the current aid recommendation. In addition, the user will have some information about the *aid recommendation itself* simply by virtue of having decided to

monitor the aid, whether or not the user chooses to collect more information by *verifying* that recommendation: (1) First, the user knows the *type* of recommendation that is involved. Many decision aids support more than one type of decision, which the user might decide to monitor. For example, an attack planning aid may recommend a battle position as well as a route to that position. A target identification aid might classify contacts and also prioritize them for engagement. The user may trust the aid more on some of these matters than on others, and this trust will influence decisions about verifying the conclusions. (2) Second, the user may be aware of the *content* of the aid conclusion or recommendation, and this too can influence verification decisions. For example, the user may trust identifications of contacts as friends (since they are based on reliable Identification-Friend-or-Foe (IFF) procedures), but not trust identifications of contacts as foes (since friends may stray from designated areas or turn off their IFF transponders). (3) Finally, the user may be aware of the aid's own reported *confidence* level, or its *explanation* of its reasoning in arriving at the conclusion. These reports, too, (if the user trusts them!) may influence decisions about whether or not to verify the conclusion.

Verification includes a number of different activities, such as checking the aid's reasoning, examining the aid's conclusion against evidence known to the user but not to the aid, or attempting to find (or create) a better alternative. Verification is not usually a once-and-for-all decision. More typically, it is an iterative process, in which users continue thinking and collecting information about aid recommendations until they resolve the uncertainty, the priority of the issue decreases, or the cost of delay grows unacceptable. If the user does decide to verify an aid recommendation by collecting more information, that new information will then be available to influence subsequent decisions to continue or not to continue verifying. If the user chooses to continue verifying, the user may consult more of the available evidence or try out a different verification strategy. The process should end when there is little remaining uncertainty, when other priorities demand the decision maker's time, or when action cannot be delayed.

Decision Trees for Verification

The process of discovering new information or insights during verification can be pictured as an event tree, in which the user's trust in the aid evolves as new observations are made. The verification decision at any given time is based on the trust in the aid that the user has at that time. It is illuminating to incorporate verification decisions within the tree, as events under the control of the user. A tree that includes both chance events and decisions is called a decision tree (Raiffa, 1968; Shafer, 1997).

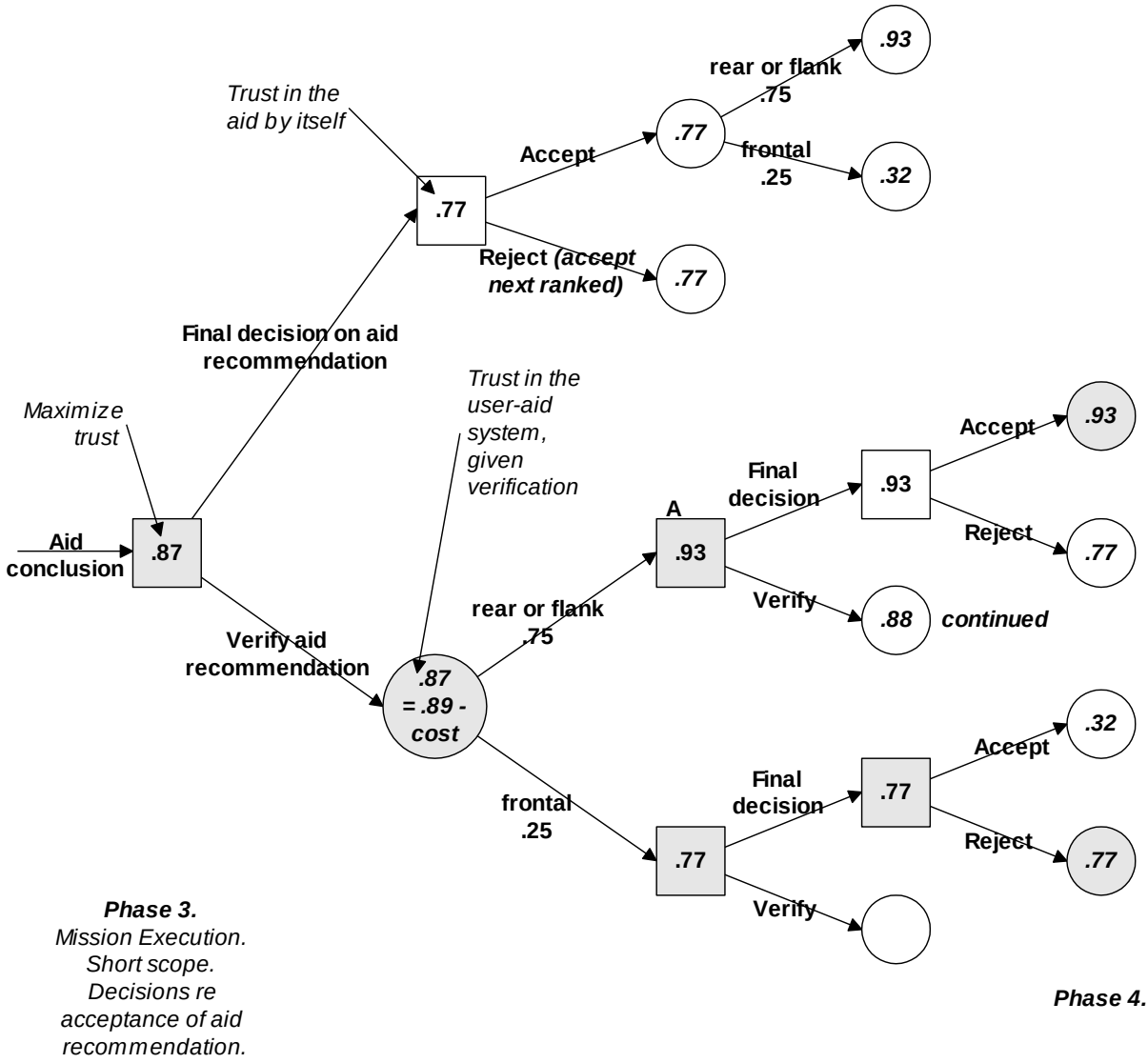

Figure 14. A decision tree showing a verification decision and the subsequent decision to accept, continue to verify or reject the aid's recommendation. Shading indicates the part of the tree that the user may traverse, depending on chance events (circular nodes) and decisions (square nodes).

Figure 14 is a decision tree with two verification decisions: an initial decision of whether or not to verify an aid recommendation, and, if the user chooses to verify, a subsequent revisiting of that decision. This example is based on Figure 6: A user of the CBPR aid has been assigned a desert attack mission, and at the beginning of Phase 3 the aid has recommended a combat battle position. The user must first decide whether to accept this recommendation at once, reject it at once, or verify it by collecting more information.[9] In the previous chapter, we discussed completeness and the associated property of resolution in the user's knowledge regarding aid effectiveness. We now see that completeness and resolution are shaped by the user's own decisions. In particular, verification decisions in Phase 3 determine the completeness of the subtree that the user will traverse during the remainder of that phase. For example, if users decide to verify the conclusion in Figure 14, they will traverse a subtree that contains branches for

[9] We make several simplifying assumptions in Figure 14: First, we omit other final decision options, such as modifying the aid's recommended battle position by shifting it slightly or expanding or contracting its size. Second, only one verification option is considered: observing the relation of the recommended battle position to the likely enemy avenues of approach. (We will extend the example on this point in Figure 15.) Third, we assume that if users reject the top-ranked aid recommendation, they will accept the next highest ranked aid recommendation. As we shall see, all of these assumptions can easily be relaxed in a more elaborate decision tree.

different angles of attack (as in the more complete tree of Figure 6). Angle of attack information will then be included in the Grounds of subsequent judgments of trust, and the subsequent accept/reject decision will be based on this information. On the other hand, if users choose not to verify the aid's conclusion, they face a subtree that is missing branches for angle of attack (like the incomplete tree shown in Figure 8), grounds for trust will not include angle of attack information, and the information will not be available for the accept/reject decision (although, like rotorwash, angle of attack may become known *later*, after the recommendation is executed in Phase 4). When the user chooses not to verify, the accept/reject/modify decision will be based on *average* trust, aggregated across the various angle of attack possibilities, rather than on specific knowledge. One important goal of reliance decisions is to improve the *resolution* of the judgments upon which subsequent decisions are based.

The numbers at each node in Figure 14 are calculated in much the same way as in the previous chapter.[10] However, these numbers do not represent the user's trust in the decision aid alone. In a decision tree, they represent the user's trust, at that time, in the *overall user-decision aid system given any reliance choices already made by the user*. Trust represents the expected, or average, chance of successful collaborative user-aid performance in the future, conditional on such collaboration in the past.

Is it worthwhile for the user to verify the aid's recommendation? The answer is surprisingly simple, and involves a comparison of what is, in effect, *trust in the aid by itself* with *trust in the overall user-aid system given that the user will verify*. The user chooses the path through the event tree that gives the best chance of ending up with successful attack. We know from Figure 6 that the user's trust in the aid by itself before learning angle of attack is .77. This number reappears in Figure 14 as trust in the overall user-aid system given that the user chooses *not* to verify. In Figure 14, trust in the user-aid system given that the user *does* verify is .89 minus the costs of verification. Such costs may include heightened risk, for example, of being targeted by the enemy, or loss of opportunity to perform other important tasks.[11] Suppose the user estimates such risk as no greater than a 2% reduction in chance of successful attack. Since the expected success of the user-aid system is greater if the user verifies the conclusion (.89 - .02 = .87) than if the user does not (.77), the user should verify. *Users can make reliance choices by maximizing trust.* (See Appendix A for a more formal development of these ideas.)

We now elaborate the example in Figure 14 to illustrate the iterative character of verification decisions, and also to demonstrate a case in which collecting more information is *not* worthwhile. Incorporating iterative verification possibilities will increase the expected value of verification, but will also, of course, increase its expected costs.

[10] However, trust is calculated differently at the two kinds of nodes in Figure 14 At the circular, chance nodes, trust is the probabilistic average of the possibilities branching to the right (as in the previous chapter). At the square decision nodes, trust is equal to the trust at the right-hand node that is chosen by the user. We assume that the user will adopt the option that offers the highest chance of success, i.e., that the user will maximize trust. Trust at a decision node is thus the maximum of the trust values that the user sees looking toward the nodes to the right. (Some potential problems with this approach, and remedies, are discussed in Cohen & Freeling, 1981.)

[11] This formulation assumes that cost is measured in the same units as decision-aid user performance. In other words, we measure the cost of delay in terms of the decrements it causes in chance of successful attack.

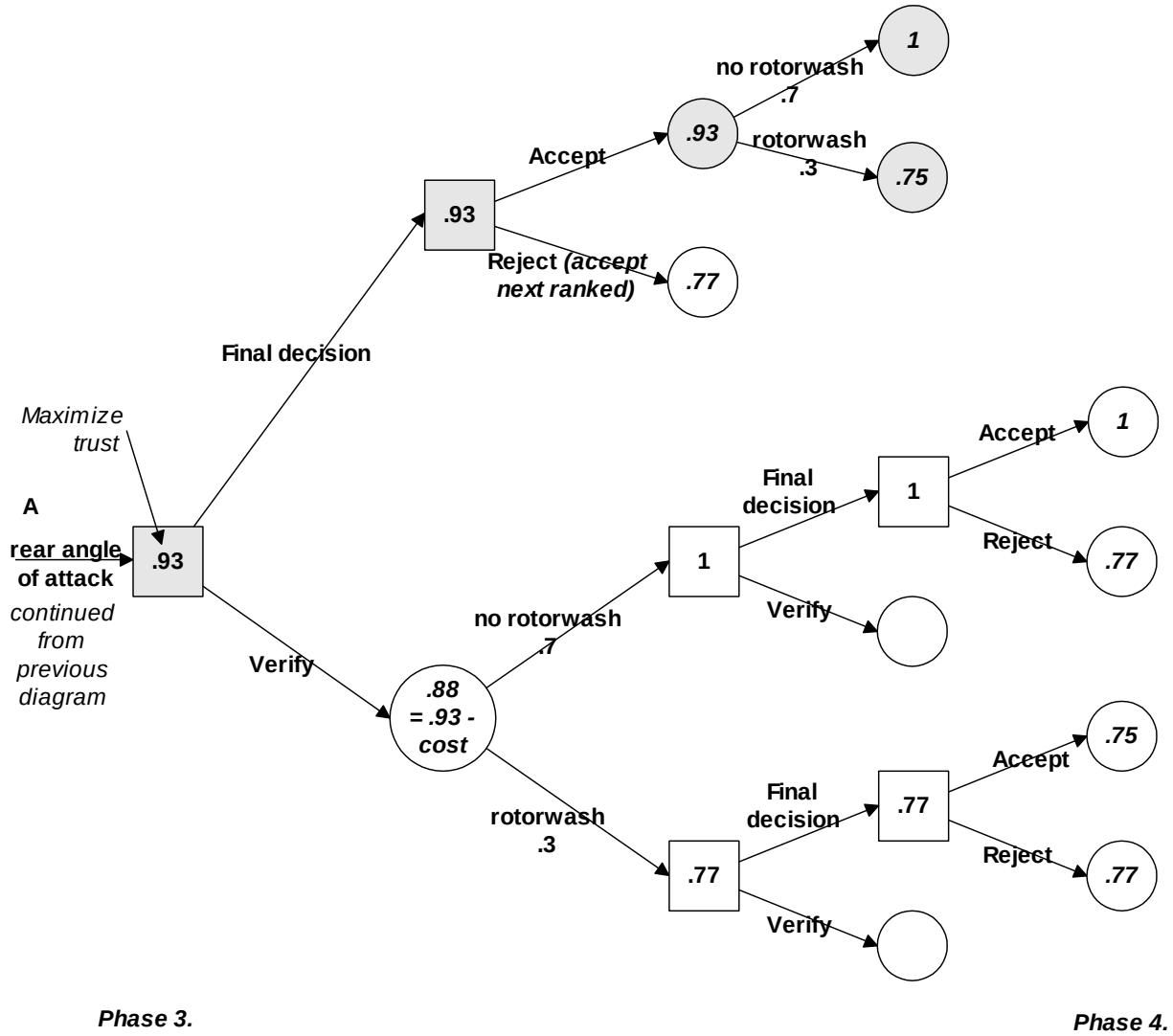

Figure 15. Decision on whether to continue verifying by examining rotorwash. Node marked "A" in this figure corresponds to node marked "A" in previous figure.

Figure 15 extends a branch in Figure 14 that represents the option to continue verifying. Suppose the user decided to examine the angle of attack of a recommended battle position and found that it involved a rear attack. Now the user must decide if it is worthwhile to also examine rotorwash. To do this, suppose the user would have to delay the attack long enough to visit the site, at a cost of 5% chance of attack success. Figure 15 indicates that this additional verification is not worthwhile. If verification were without cost, it would still have virtually no effect (within rounding error) on chance of success. But when the 5% cost is factored in, taking the time to look at rotorwash yields a net reduction in chance of success, or overall trust. According to this model, verification should come to an end after angle of attack has been determined.[12]

Value of Verification Information

The user in Figure 14 appears to benefit significantly by verifying the aid's recommendation. It is illuminating to consider why this benefit occurs. There are three basic requirements:

- *Uncertainty:* The information to be collected must be capable of *changing a subsequent decision*, such as whether or not to accept the aid's recommendation. This change can only occur as a result of higher

[12] Recall that the probabilities in the example are illustrative only, and do not reflect recommendations for any real situation. In particular, inspecting a recommended battle position may reveal more information than simply rotorwash.

expected resolution in the trust judgements based on the new information. If resolution is already perfect (probabilities equal 0 or 1), new information has no value.

- *Stakes:* Change in a subsequent decision must be capable of *changing payoffs*, e.g., chance of successful attack. The higher the cost of an error, the more valuable verification becomes.
- *Time:* The cost of delay required to verify must be outweighed by the first two factors.

All of these conditions apply in this example. First, *obtaining information about angle of attack can change the user's choice of battle position*. Verification can result in either of two relevant outcomes: The user may observe that the recommended battle position involves a rear or flanking angle of attack, or that it involves a frontal angle of attack. In the first case the user will accept the recommendation, and the expected success is .93. In the second case, if the user were to accept the recommendation, expected success plummets to .32. Instead, the user will reject the recommendation and (we will assume) look at the next highest ranked aid recommendation. In other words, verification can, in this example, lead to discovery of an error important enough to shift the user's acceptance decision. The chance that this will happen is .25 (i.e., the probability that the aid has selected a frontal angle of attack in desert terrain), which is by no means negligible.[13]

Second, by rejecting a frontal angle of attack, *the user increases the chance of a good outcome*. Trust grows from .32 (if the user were to go ahead and adopt the recommended frontal battle position) to .77 (the overall chance that the next top ranking aid recommendation will be acceptable). This represents an improvement of .45 in the chance of successful battle position selection.

Third, the first two factors outweigh the cost of delay. The expected, or average, improvement due to verification in this situation is the product of the first two factors, i.e., the chance of an observation that changes behavior and the benefit realized by changing behavior, minus costs: (.25)(.77 - .32) - .02 = .09. This corresponds (within rounding error) to the measure we discussed earlier: the difference between trust in the overall user-aid system given that the user chooses to verify (.87) and trust given that the user does not chose to verify (.77). By deciding to verify —even before collecting the information about angle of attack — the user manages to increase the expected success of joint user-aid battle position selection (i.e., trust) by 12% of its original value.

It is also easy to see why verification of rotorwash is not justified in Figure 15, by looking at the same three requirements. First, can information regarding rotorwash change the user's decision? It can: If there is rotorwash, the user will switch from acceptance to rejection of the recommended battle position. The chance that this will happen is .3. Second, will the change in decision cause a change in payoffs? Again, the answer is yes, but the effect is very small. There is a .75 chance of success from a battle position with rotorwash (given a rear angle of attack). This rises only to .77 if the user chooses to look at other battle position recommendations. The combination of these two factors – (.3) (.77-.75) = .006 – is vastly outweighed by the third factor, the assumed .05 risk in obtaining the information.

A convenient tool for putting these ideas together, and for building benchmark models of reliance decisions, is the decision theoretic concept of *value of information* (VOI) (Cohen & Freeling, 1981; Raiffa & Schlaifer, 1961;

---

[13] Most of the numbers in Figure 14 parallel those in the example in Figure 6: for example, the .75 / .25 odds of a rear or flanking angle of attack in desert terrain, the .93 chance of success given a rear or flanking angle of attack, the .32 chance of success given a frontal angle of attack, and the .77 chance of success prior to obtaining angle of attack information. The trust associated with deciding to verify is simply the probability-weighted average of the trust associated with its possible outcomes: i.e., (.75) (.93) + (.25) (.77) = .89.

For the purposes of this example, we have assigned a value of .77 to the option of rejecting the aid recommendation. As noted, this assumes that when rejecting the aid's initial recommendation, the user will accept the aid's next-ranked recommendation, and that its chance of success is equal to the expectation regarding the aid's initial recommendation. This is only an approximation, to keep the model simple. First, if the decision were binary (e.g., identification of a target as hostile or not-hostile), rejecting the aid's initial conclusion would be equivalent to accepting its negation; trust in the negation equals one minus trust in the rejected conclusion (in this example, 1 - .32 = .68). In non-binary, open-ended cases, such as selection of a battle position, the approximation may be conservative. For example, the value could be higher if the user plans to continue verifying options in the expectation of eventually finding a battle position that is acceptable to the aid *and* has a rear or flanking angle. If the cost of such continued verification is low enough, this might increase the expected value of the verification strategy, making it even more preferable to not verifying. On the other hand, if the aid expresses very low confidence in its own subsequent recommendations, the user might revert to a manual strategy for this decision, incurring a higher cost in time and risk. By the same token, prior observations regarding the likely number of good battle positions in the area might affect the expected value of looking for another battle position manually.

LaValle, 1968; see Appendix A for more details). A simple formula for value of information, as applied to verification decisions by aid users, is the following:

Value of verification information = Sum over all observational outcomes that could change the user's subsequent decision

*[probability of the observational outcome * change in trust due to the change in decision – cost of time spent making the observation]*

The user should verify the aid's recommendation if this value is greater than zero. Value of information is a significant improvement over other information measures, such as entropy reduction, which measure the sheer quantity of information without taking into account the reason why information may be of value, i.e., its actual role to support decision making. And it is better motivated and simpler than the large number of rather vague measures typically used in information management system research, such as *completeness, precision, accuracy, relevance, timeliness, clarity, and readability* (see Cohen & Freeling, 1981, for discussion). None of these other frameworks can explain why, in our example, the user should verify angle of attack but not rotorwash.

Table 7 summarizes some dimensions of generality of the Value of Information model. For example, in Figure 14 and Figure 15 we limited the final options available to the user to accepting the aid's recommendation, or rejecting it (and accepting the aid's next ranked recommendation). However, value of information is quite general in this respect. Decision aid users may *modify* an aid recommendation in any number of ways. They may even adopt a verification strategy in which they search for or create new options, as shown in Appendix A.

Table 7. Generalizations of the Combat Battle Position Recommendation example within the value of information framework. (Letters in parentheses in the left column refer to the formal notation in Appendix A.)

Element of model	**CBPR Example**	**More General Case**
Observations that can be made during verification *(v)*	angle of attack, rotorwash	more observable features; methods for modifying recommended options; strategies for finding or generating new alternatives
Possible outcomes of observation *(z)*	rear/flanking vs. frontal	more possible observational outcomes per feature (e.g., the number of engagement positions within a battle position; discovery of new battle positions; the number of possible vehicle shapes on a FLIR image; finding a better attack plan)
Options in subsequent decision *(a)*	accept vs. reject aid recommendation	more subsequent decision options, depending on the type of conclusion: e.g., accept, accept another alternative, modify in various ways
States of world that influence utility of outcomes *(s)*	successful battle position vs. unsuccessful	more states of the world that affect outcome utility (e.g., kill high-valued enemy assets, kill low-valued enemy assets, kill friendlies)

Let us briefly consider another, somewhat more complex example, which illustrates some of these dimensions of generality. Suppose a target identification decision aid does not incorporate the overall shape of a vehicle in its algorithms, but focuses instead on details such as the number of wheels, presence of FLIR hotspots, etc. A user could decide to verify an aid conclusion regarding target identity by quickly checking the overall silhouette of a vehicle image (Cohen, Thompson, & Freeman, 1997). The outcomes of this observation consist of a large space of possible shapes (parameter z in Appendix A), varying along dimensions such as "boat-like," "truck-like," etc. The user may then select from a set of options (parameter a in Appendix A): accept the aid's identification conclusion, modify it in varying degrees and then accept the modification (e.g., changing the identification as T-62 to another tank, e.g., T-55, which differs subtly in shape, or to a truck, such as KRAZ, which differs more significantly in shape), continue to verify the conclusion by examining other features of the image (parameter v Appendix A), or take another possible classification (e.g., what if it is a friendly truck?) and try to verify that. The decision of whether or not to verify will be influenced by a utility function that distinguishes costs of different kinds of errors (e.g., confusing an enemy tank with a friendly tank, or an enemy tank with an enemy truck), and different degrees of success and failure (parameter s in Appendix A). For example, a greater cost will be assigned to misidentifying a friend as a foe (leading to possible fratricide) than to misidentifying an enemy truck as an enemy tank.

In the previous chapter, we defined trust in terms of probabilities: the predicted chance of a successful aid recommendation. An expanded definition of trust has been presented in this chapter, which is more general in two respects: First, it expands to include the user's own potential role in interacting with the decision aid. Second, as we emphasized in the last paragraph, it incorporates utility, or preference. Trust can be regarded as the *expected utility* of the aid recommendation, or the user's modification of that recommendation, which includes, but goes beyond, its *chance of success*. The effect of this new definition is simply to weight successful and unsuccessful conclusions by their importance. Thus, an aid (or an aid-user collaboration) that is successful where it counts and unsuccessful where it doesn't count, earns more trust than an aid (or a user modification policy) that is successful where it doesn't matter and unsuccessful where it does. As one might expect, this definition yields some reasonable predictions: For example, *experienced users should tolerate less inaccuracy in a target identification aid and be more likely to verify its conclusions, in close air support missions (where friendlies are present and a mistake can be costly) than in deep interdiction missions (where friendlies are not present)*. More generally, there is no single level of accuracy (i.e., trust in the sense of chance of success), such as 80% or 95%, that is necessary and sufficient across all situations for acceptance of a decision aid, since the costs of errors can vary considerably.[14]

Dynamic Constraints on Verification

Despite their generality, measures based on the value of information have limitations. Primarily, these limitations flow from the requirement that the possible observational consequences of verification be specified explicitly in advance (Cohen & Freeling, 1981). This was not an unreasonable requirement for the purpose of training quick recognition of standard patterns. However, the advantage of interactive over automated systems may be the human ability to handle novel and unexpected situations. In these cases, the possible results of human intervention may not be known ahead of time. There are several, closely related problems:

1. *Visual recognition.* The verification process may be very straightforward in some cases, yet the potential observations cannot be anticipated. For example, the user of a target identification aid can verify identification of an image as a hostile tank simply by looking at the image, yet it might be very difficult to specify in advance all the relevant details that the user might see. (For an application of the model in that domain, see Cohen, Thompson, & Freeman, 1997.)
2. *Incomplete mental models.* More generally, a decision aid user may not have a detailed event tree predicting how trust will evolve as a function of future observations. For example, a user's mental model of aid performance may be based on recognition of a particular situation rather than prediction, and thus be closer to Figure 8 than to Figure 6. Even though such users do not have an explicit awareness of what they are looking for, it may still be very much worthwhile for them to "take a look" at the aid's recommendation before implementing it.
3. *Critical thinking.* The verification process itself may be less straightforward in some situations. For example, conflict between an aid's recommendation and their own or others' conclusions, may prompt a process of critical thinking, in which users look for an explanation of the differing recommendations. Resolution of the conflict may take the form of discovering unreliable assumptions that were implicit in the soldiers' conclusions or the aid's. It is virtually impossible to make all assumptions explicit in advance in an event tree. Key assumptions may come into focus only when they lead to problems, such as conflicting recommendations (Cohen, Freeman, & Thompson, 1997).
4. *Novel situations.* More generally, new issues to investigate may spring up as a result of unique or unusual circumstances, or due to the pattern of ongoing verification results. Just as novel situations may not be anticipated by the designer of a decision aid, so they may not be anticipated by the training designer.
5. *Information interdependence.* The value of one piece of information may be very low when considered by itself, but high when considered in the context of other observations that might subsequently be made. For example, even after discovering that a recommended battle position is unacceptable in some respect (e.g., rotorwash), a user may choose not to reject it. However, discovering a second or third problem (e.g., bad angle of attack, no room for multiple firing positions) might be enough to trigger search for a new battle position. The examination of rotorwash turned out to be worthwhile because it eventually helped tipped the balance against the recommended battle position. In intelligence assessment, the significance of one piece of information may be unclear until other pieces of the puzzle

[14] In the angle of attack example, we did not distinguish degrees of success and failure. We adopted a simple utility function that assigned utility of 1.0 to accepting a successful battle position, and utility of zero to selecting an unsuccessful battle position. In that case, the expected utility of the aid recommendation was the same as the probability of its success.

are obtained. The decision of whether or not to verify the first feature must therefore take into account the possibility of continuing the verification process to include the other features.

Fortunately, these difficulties can be surmounted without giving up the essence of the value of information approach. We will describe a simple framework for deriving benchmark models of verification performance, without specifying all possible observations. The framework can, therefore, be applied to situations where previously learned or explicitly identified patterns may be insufficient to guide decisions about user-aid interaction. This framework will thus apply even when verification involves visual recognition of unanticipated patterns, incomplete mental models of aid performance, critical thinking that ferrets out hidden assumptions, creative problem solving in novel situations, and interlocking or reinforcing pieces of information. The solution is to derive necessary conditions, or *constraints*, that must be satisfied if any verification at all is to be of value. If the situation does not satisfy these constraints, verification cannot be worthwhile, regardless of the number of unmodeled potential observations and insights. These constraints need not be static, but may change dynamically as a the situation itself evolves.

Rather than attempting to model individually all the observations that could be made during verification, we will assume that *perfect information* is obtained. A simplified model can be obtained by assuming that verification will produce observations that are perfectly correlated with outcomes that determine the appropriate reliance decision. If verification is not worthwhile under this assumption, then it cannot be worthwhile under more limited conditions. It turns out that these constraints can be expressed relatively simply, in terms of current trust in the aid by itself, the costs of verification, and the potential loss that might be avoided by verification (see Appendix A for a derivation). In particular, users should accept an aid recommendation without verification if:

trust > 1 - cost of verification / the cost of incorrectly accepting the aid recommendation

This constraint is expressed as a ratio of costs. However, the denominator can be rephrased, equivalently, in terms of potential benefits. Verification is not appropriate if:

trust > 1 – costs of verification/ expected benefits if the aid recommendation is wrong

The denominator on the right is the "upside" of verification: It refers to the additional utility that will be realized by the discovery of a better option *if* the aid's recommendation is in fact incorrect. We shall refer to this as *conditional benefit*. The numerator, by contrast, is the cost in time or risk that will be incurred by verification whether it turns up anything interesting or not.

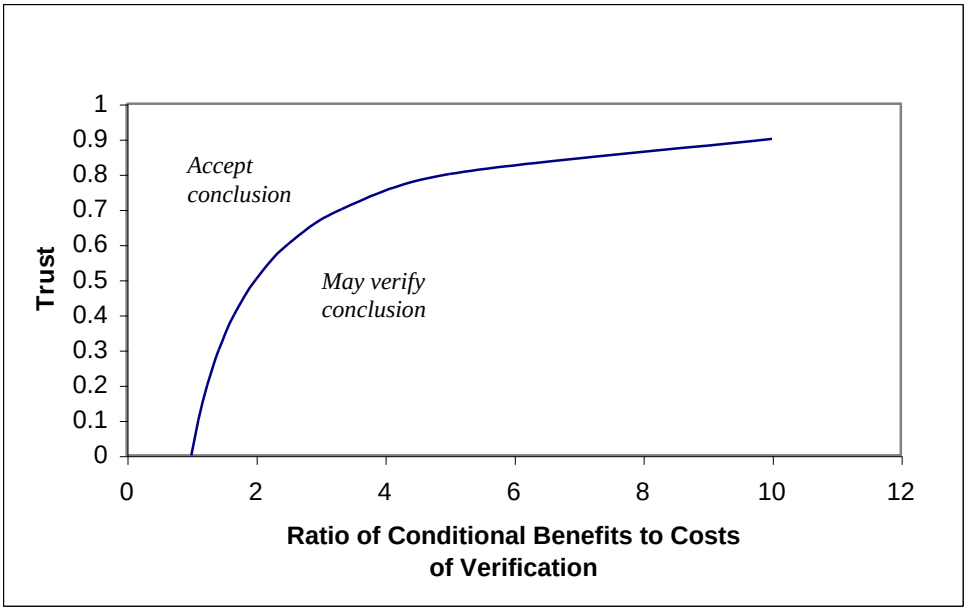

Figure 16. The ratio of benefits (given that the current recommendation is wrong) to costs that is required to justify verification, as a function of trust in the aid's recommendation.

Figure 16 shows how the upper bound on trust varies with the ratio of conditional benefits to costs of verification. For a given level of trust in the aid's recommendation, the chart shows how much conditional benefit relative to cost the user should require from a verification strategy. If trust is 75%, Figure 16 shows that the potential benefit of verification must be at least four times the cost for verification to be justified. If trust is 90%, the potential benefit must be at least 10 times the cost, i.e., the user needs to have a much more efficient approach to verification for it to

be worthwhile. Notice also that unless the potential benefit of verification exceeds the cost (i.e., ratio > 1), no amount of distrust can justify verification.

Types of Verification Strategies

Verification is defined as an information collection process that can change the user's expected utility for options. The type of decision and the number of options facing the decision aid user, therefore, influence the types of verification strategies that are available. And the types of verification strategies that are available, in turn, affect the way in which dynamic constraints apply.

It will be useful to introduce a classification of aid recommendations into three types, each associated with different possibilities for verification. We will distinguish *binary* recommendations and two types of *non-binary* recommendations*: close-ended* and *open-ended.*

(1) An important, but limited class of aid recommendations is *binary*, for example, a recommendation to engage or not engage a target, or an identification of a contact as friend vs. foe. In these cases, if the aid's recommendation A is wrong, the correct conclusion must be not-A. Modification of the original conclusion is therefore equivalent to rejecting it and accepting its negation. Thus, the choices for final dispensation available to the user are limited to (i) accept A, or (ii) modify A = reject A = accept not-A. Moreover, verification of A is equivalent to verification of not-A, since the same observations are relevant to each; and since there are only two relevant alternatives, there is no need for the verification process to find or generate new ones. The only verification strategy is to verify A / not-A – though there may be different verification options regarding which features to investigate first.

(2) A second class of aid recommendations is *close-ended*, in which the answer comes from a prespecified set of alternatives, e.g., A, B, C, or D. Examples are classification of a target according to type, e.g., tank, armored personnel carrier, truck, or jeep, or management of sensor modes by selecting from a limited set of options. Here, users have more choices for final dispensation: for example, (i) accept the aid recommendation A, (ii) modify A by choosing alternative B, (iii) modify A by choosing C, (iv) or modify A by choosing D. Moreover, for a non-binary, close-ended conclusion, verification of one alternative is not the same as verifying another, since somewhat different observations may be required for establishing not-A, for example, versus narrowing down the answer to C. Thus, broad verification strategies include: verifying A, verifying B, verifying C, and verifying D.

(3) The third class of aid recommendations is *open-ended*, in which the alternatives are not all specified in advance. The example of selecting a battle position for attack illustrates this case, because there is no pre-existing list of all the potential battle positions from which the user or the aid can select. Other examples include relatively ill-defined tasks such as the creation of an overall concept of operations for the mission, or the assignment of a large number of specific units or platforms, with different ordnance packages, to a large number of possible targets or

locations. For an open-ended decision, users have many more choices for final dispensation of the aid recommendation: (i) Modifying the aid's recommendation may involve selecting a new alternative from a set of currently known options (such as B, C, D, or E); an example would be shifting attention to the aid's next ranked recommendation, and then to the next-ranked one after that. (ii) In addition, modification options may include small adjustments, such as altering the size or position of a battle position, changing the ordnance mix, or adding a second battle position to the one recommended. (iii) Finally, the user might choose an option not recommended by the aid at all. Verification options are correspondingly richer for open-ended conclusions: Users may verify A, verify one of the other known alternatives (e.g., the next-ranked aid recommendation); they may adopt some strategy for modifying one or more of the known alternatives; or they may attempt to find or generate additional alternatives whose identify and even existence is not presently known. An example of the latter is reverting to manual performance and searching for alternative battle positions on a paper map. For open-ended decisions, therefore, the cost of verification in time and risk will sometimes be equivalent to, or higher than, the cost of manual performance. As we shall see, the open-ended character of a conclusion tends to make it less likely that a user will reject the currently known options, unless trust in them is unusually low.

As this discussion implies, we can classify verification strategies according to the scope of their influence on user options:

(1) V_1 is a class of relatively low-cost strategies that collect information about known options only (e.g., battle positions recommended by the aid). The maximum improvement that V_1 can deliver is the difference in expected utility between the currently preferred option and the best of the known options. V_1 cannot reveal significantly better options than those already known.

(2) V_2 is a class of strategies that modifies known options (e.g., shifting the size or location of a battle position slightly). This is more costly in time and effort, but may deliver a somewhat greater benefit. The maximum improvement is the difference in expected utility between the currently preferred option and the best of all the improvements that might be made to known options.

(3) V_3 is a class of strategies that discovers or creates new options (e.g., finding battle positions on a paper map).[15] It is associated with both the highest cost and the highest potential benefit. The limit on improvement is determined only by the best possible option that can be found or constructed in the task.

We will first consider a special form that the constraints on verification take when the user's decision is binary. We will then move on to the more general case, in which the user's options are non-binary.

Binary Decisions

If the aid conclusion is binary (e.g., classification of a contact as friend *or* foe), we can fully characterize the user's decision by two constraints on the appropriateness of verification strategy V_1: an upper bound on trust (above which users should simply accept the recommendation) and an upper bound on distrust (above which the users should simply accept the *negation* of the aid recommendation). An upper bound on distrust is, of course, equivalent to a lower bound on trust. Users may choose to verify if neither of the two constraints on trust is the case, i.e.:

1 - cost of verification / the cost of incorrectly accepting the aid's recommendation > trust > cost of verification / the cost of incorrectly rejecting the aid's recommendation

Trust, in these equations, refers to the chance of a successful aid recommendation. This narrow definition, which was introduced in the last chapter, is appropriate here since the costs of different kinds of errors have been separated out and represented as constraints. Appendix A describes the derivation of these constraints in more detail.

Figure 17 represents a benchmark model for a binary decision based on these constraints. Trust in the aid (the probability that a particular decision aid conclusion is correct) is shown on the vertical axis, ranging from no trust (0) to complete trust (1.0). Time is plotted along the horizontal axis. Thus, the long-dotted line shows that confidence in the aid begins relatively low; in fact, if further verification was not possible, this user would choose to reject the aid conclusion. However, trust increases in this example with time spent verifying the conclusion. This might happen, for example, if the angle of attack of a recommended battle position is discovered to be flanking, it is then observed that there is lots of room for other aircraft in the recommended position, and so on. Of course, confidence could also have declined as new evidence was considered, for example, if the angle of attack turned out to be frontal, there was insufficient room for other aircraft, and so on. As described in the previous chapter, the evolution of trust depends on the path through the event tree of potential observations that is realized as the user collects information.

[15] As discussed in Appendix A, we assume that expected utility of options not yet considered by the user is zero. The process of discovering or creating new options can thus be modeled as a form of verification that changes their expected utility to a positive or negative value.

At any point in time, the vertical dimension is divided into two or three regions. If trust in the aid's conclusion falls in the upper region, the user should simply accept the conclusion (e.g., engage the target), without taking further time for verification. If trust in the aid's conclusion falls in the lower region, the user should reject the aid's conclusion without taking further time. (For example, a target identification aid concludes that a vehicle is an enemy tank, but the user is reasonably sure based on visual identification that the target is a friendly.) If trust is neither high nor low, but falls in the intermediate region, then it may be worthwhile for the user to take more time to decide what to do. In Figure 17, the user's initial level of trust warrants further verification of the aid's conclusion. After a while, however, the user's confidence in the aid has increased enough to enter the upper region, where its conclusion should be accepted. At this point, the user should stop thinking and act.

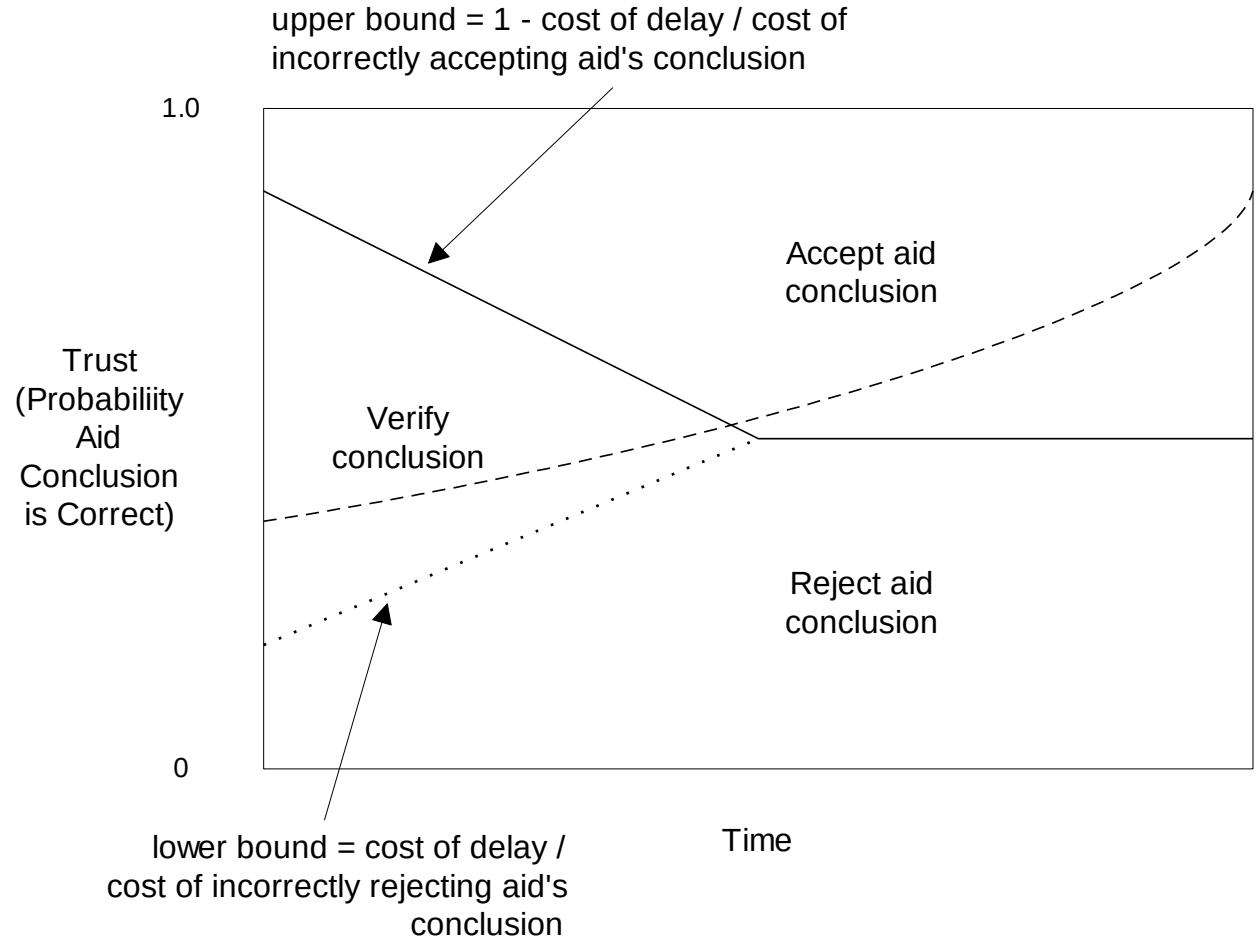

Figure 17. Benchmark model for deciding when to accept, reject, or take time to verify a decision aid's conclusion. Trust is represented by the long-dotted line.

What determines reliance decisions in this model? This surprisingly powerful representation has only three key variables: uncertainty, time stress, and stakes.

1. *Uncertainty* pertains principally to the *resolution* of the trust assessment, i.e., the proximity to zero or one of the probabilities discriminated by the user. The less resolution in the user's assessment of trust, the more likely that a calibrated assessment will fall in the middle region of Figure 17, and the user will tend to utilize more time before making a decision. As we noted in the discussion of APT (Figure 2), the resolution of a trust assessment is influenced by the *completeness* of the user's knowledge of conditions that affect system performance. The more complete the knowledge of relevant features of the domain, situation, task, and system, and the more reliably these features are observed on a given occasion, the closer the calibrated trust assessments will come to zero or one. We now see an important implication of the connection between completeness and resolution. *Training that improves a user's knowledge (prior to Phase 3) of features that predict aid performance will reduce the amount of time the user needs to spend verifying the system (in Phase 3).* Informed users will be able to assess the value of aid recommendations more quickly.

2. Time stress is represented by the cost-of-delay parameter in the equations determining the upper and lower bounds (see Figure 17). When the cost of delay is great, action is more imperative, even with relatively low-resolution levels of trust. The cost of delay need not be constant, but may itself be a function of time. As time stress increases, the upper boundary moves down and the lower boundary moves up, reducing the size of the intermediate region where verification might be appropriate. For example, the risk of being targeted by an enemy may increase with the time spent unmasked, e.g., to verify the enemy's identity. Time stress can also increase due to neglect of other tasks, which grow increasingly urgent. Figure 17 illustrates such a case, in which the cost of each further moment of delay is higher than the one before, until finally no further delay is justified. When the upper and lower bounds meet due to increasing costs of delay, the user must act, regardless of the level of trust.

3. Stakes. The locations of the boundaries in Figure 17 depend on the relative costs of being in each of the three regions. We have already considered the cost associated with being in the middle region; time stress affects the upper and lower bounds symmetrically, driving them closer together as it increases. By contrast, there are two different kinds of stakes, corresponding to the costs of mistakenly accepting or rejecting the aid's conclusion, respectively. These two kinds of costs affect the two bounds independently. To think about stakes, the user simply asks, regarding whatever action he or she is about to take, *what are the consequences if I am wrong?* The more severe the consequences of a mistake, the more difficult it is to clear threshold for taking the corresponding action (i.e., to get into the upper or lower region).

The upper bound increases (and the upper region gets smaller) with the cost of incorrectly *accepting* the aid's recommendation. For example, suppose that a target identification aid recommends engagement of a contact, and the user is considering accepting this recommendation. For the sake of argument, suppose that the user and the aid are wrong and that the contact is not an appropriate target (e.g., it is a friendly vehicle or an enemy non-target). Stakes are defined simply as the average difference in the expected value of engaging such a contact and not engaging it. For example, incorrectly engaging a contact is likely to be more costly, the higher the proportion of friendlies among the non-targets. Thus, increasing the number of friendlies in the area will raise the upper bound, making it harder for trust to clear the threshold for acting on the aid's recommendation to engage.

By contrast, the lower bound decreases (and the lower region gets smaller) when there is an increase in the cost of incorrectly *rejecting* the aid's recommendation. For example, suppose again that the aid recommends engagement, but the user this time is leaning away from engaging. For the sake of argument, suppose that the user is wrong in rejecting the aid's recommendation, and that the contact is in fact an appropriate target. Stakes are defined as the average difference in value between not engaging such a contact and engaging it. (This is parallel to the definition of stakes for the upper bound, except that we now assume the contact is an appropriate target.) For example, the cost of failing to engage a target are higher the more threatening the target is to one's own platform or to other friendly assets; the cost is also higher if the value of the threatened assets is higher. Thus, a bigger threat or more valuable assets to be protected can reduce the lower bound, making it harder for distrust to clear the threshold for rejecting the aid's recommendation to engage.

Non-Binary Decisions: Comparing Verification Strategies.

A far wider range of verification strategies is available when aid recommendations are non-binary, and especially when they are open-ended (such as the identification of suitable battle positions, or the development of an attack plan). Constraints on verification such as those we have been discussing, can be used to determine which strategies might be appropriate at different levels of trust.

We previously identified three classes of verification strategies: (V_1) Collecting information about known options, (V_2) modifying known options, and (V_3) attempting to find or create new options. Constraints on verification can be used to compare these (or other) verification options with one another in terms of their costs and potential benefits. Each constraint provides an upper bound on trust for the corresponding class of strategies. If trust in the aid's recommendation exceeds the constraint for a given strategy, it is not worthwhile to pursue that strategy, although some other verification strategy (e.g., with a lower cost and/or greater potential benefit) may be justified.

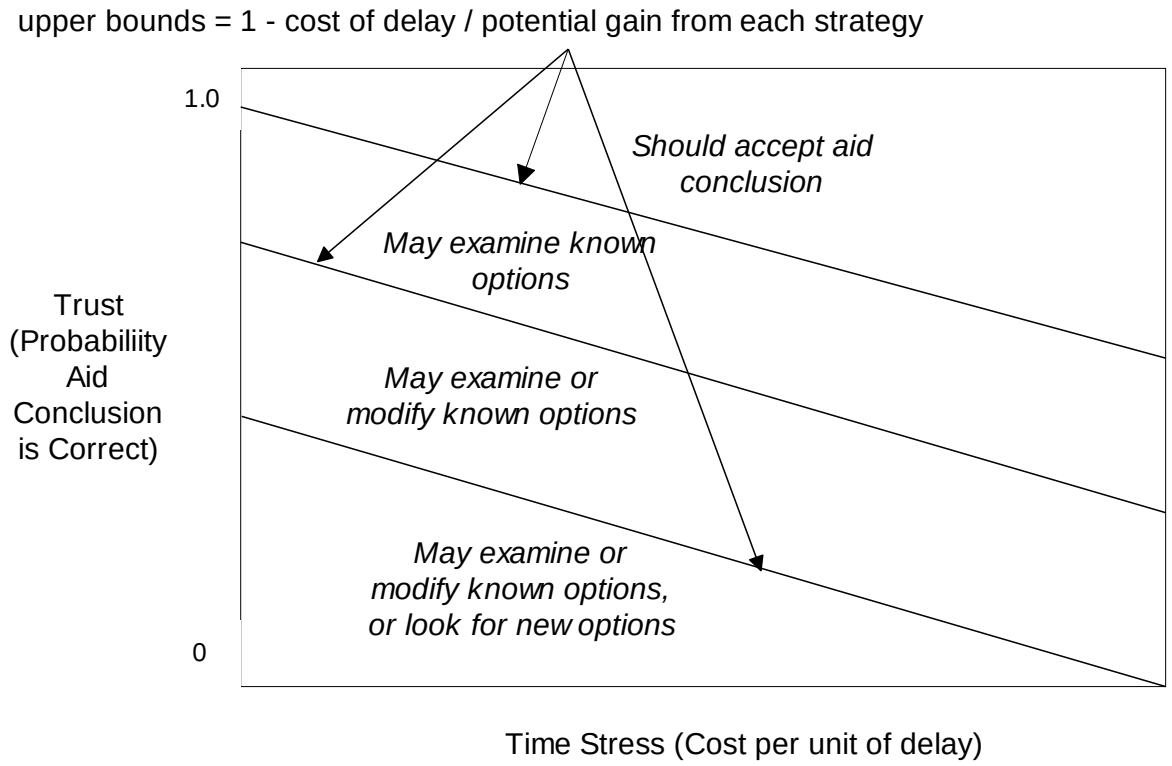

Figure 18. Illustrative regions in which the level of trust might justify different verification strategies.

It can be shown that the effect of increasing benefits is not as great as the effect of an equivalent increase in costs on the upper bound for trust (see Appendix A). Thus, as the user "upgrades" from lower cost/lower benefit verification strategies to higher cost / higher benefit strategies (e.g., the transition from V_1, to V_2 to V_3), it is likely that trust will need to clear a series of thresholds like those shown in Figure 18. That is, a growing level of distrust (or a decreasing level of trust) will be required in order to justify each upgrade in verification strategy. Users will not engage in highly effortful verification unless trust is exceptionally low.

Figure 18 also shows how the upper bound on trust for each strategy declines as a linear function of increasing time stress, or cost of delay. As the risk if delay increases, a lower level of trust is required to justify verification.

Adaptable and Adaptive Decision Aids

In the verification decision, the user functions as a supervisor of decision aid performance. The supervisory dimension of that decision is, however, limited to the single aid recommendation that is being evaluated. Other reliance decisions apply to more than one aid recommendation over a broader temporal scope, and thus take place at a higher supervisory level. These decisions, therefore, involve a more pronounced shift in the user's role from operator to supervisor. The number and nature of the supervisory decisions available to the user, however, will depend on the adaptability of the decision aid.

Figure 19 depicts a set of supervisory choices that might be available to the user of a highly adaptable decision aid in Phase 2. (It also shows some of the Phase 3 decisions to which they lead.) In this example, users make three distinguishable kinds of decisions in Phase 2:

A. Users decide whether they or the aid will have the *primary responsibility* for a given task.

B Users also determine the degree of *secondary responsibility*, i.e., the support provided by the user to the aid when the aid has primary responsibility, or by the aid to the user when the user has primary responsibility. For example, if the aid has primary responsibility, the user may choose either to allow the aid to function autonomously or to monitor its performance. The decision to monitor means that in Phase 3 the user may choose to verify the aid's conclusion; this is the verification decision that we focused on in the last section. By the same token, if users take primary responsibility for a task, they might monitor their own conclusions and verify them by comparison to the aid's solution when

appropriate in Phase 3; or they might activate the aid's monitoring capabilities, setting thresholds according to which the aid will alert them when it detects potential errors.

C Users may choose to accept the aid's default *parameter settings and rules*, or may choose to adjust them. These settings will then influence the performance of the aid in whatever mode of operation the user has selected, e.g., in autonomous performance, monitoring the user, or serving as a source of verification for the user's own solution.

Obviously, the system depicted in Figure 19 is highly adaptable by the user. A quite different design strategy requires the aid to adapt itself to the user. This approach is more characteristic of *associate* systems, or *adaptive* aids (Rouse, 1988; Morrisson & Gluckman, 1994). Figure 20 shows a decision tree in which an aid takes over some of the supervisory decisions that belonged to the user in Figure 19. Decisions under the control of the aid are represented by circular event nodes, since from the point of view of the user, they are uncertain events, even when they are designed to provide information. Users may try to predict these aid actions, in the same way that they try to deal with states of the world, like angle of attack or rotorwash.

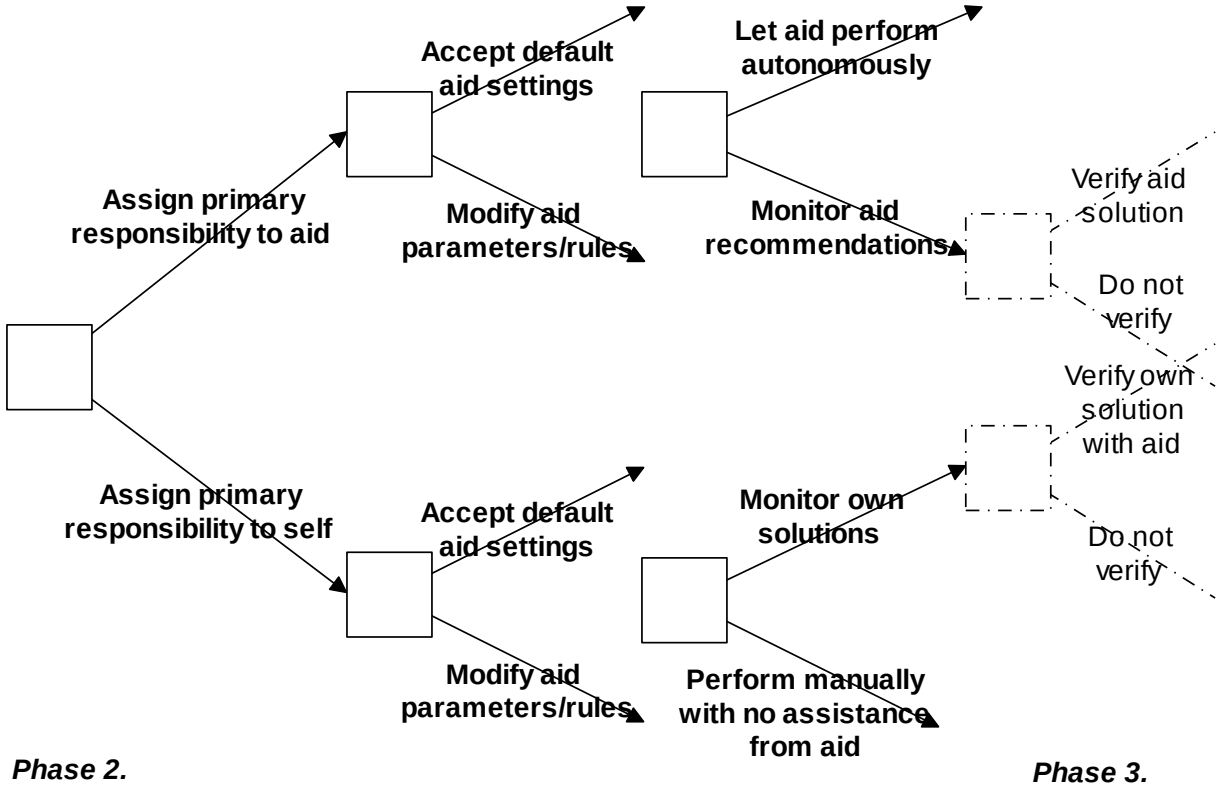

Figure 19. Illustrative supervisory decisions in Phase 2 for the user of an adaptable aid. All nodes are controlled by the user.

Figure 20 involves the same three kinds of Phase 2 supervisory decisions as Figure 19, but control over them is allocated differently. (A) The aid rather than the user makes decisions about the allocation of *primary responsibility* for each task. This may occur, for example, as a real-time dynamic reallocation of tasks as a function of user workload, skill, and other factors. (B) If the aid gives the user primary responsibility, the aid then determines the degree of *secondary responsibility* that it will assume. In particular, the aid decides if it will monitor the use or allow the user to perform without assistance. (In Phase 3, however, users will decide whether or not to verify their solutions in response to the aid's assistance.) Conversely, if the aid takes responsibility for a task, the user usually retains a supervisory role, choosing in Phase 2 whether or not to monitor aid performance. (In Phase 3, however, the aid may guide the user's attention to recommendations about which it has low confidence, and the user will decide whether or not to verify them.) (C) The user may be able to adjust aid *parameters and rules*, including some that help determine how the aid will allocate primary and secondary responsibility. Users might thus make their own preferences felt in the aid's adaptive responses.

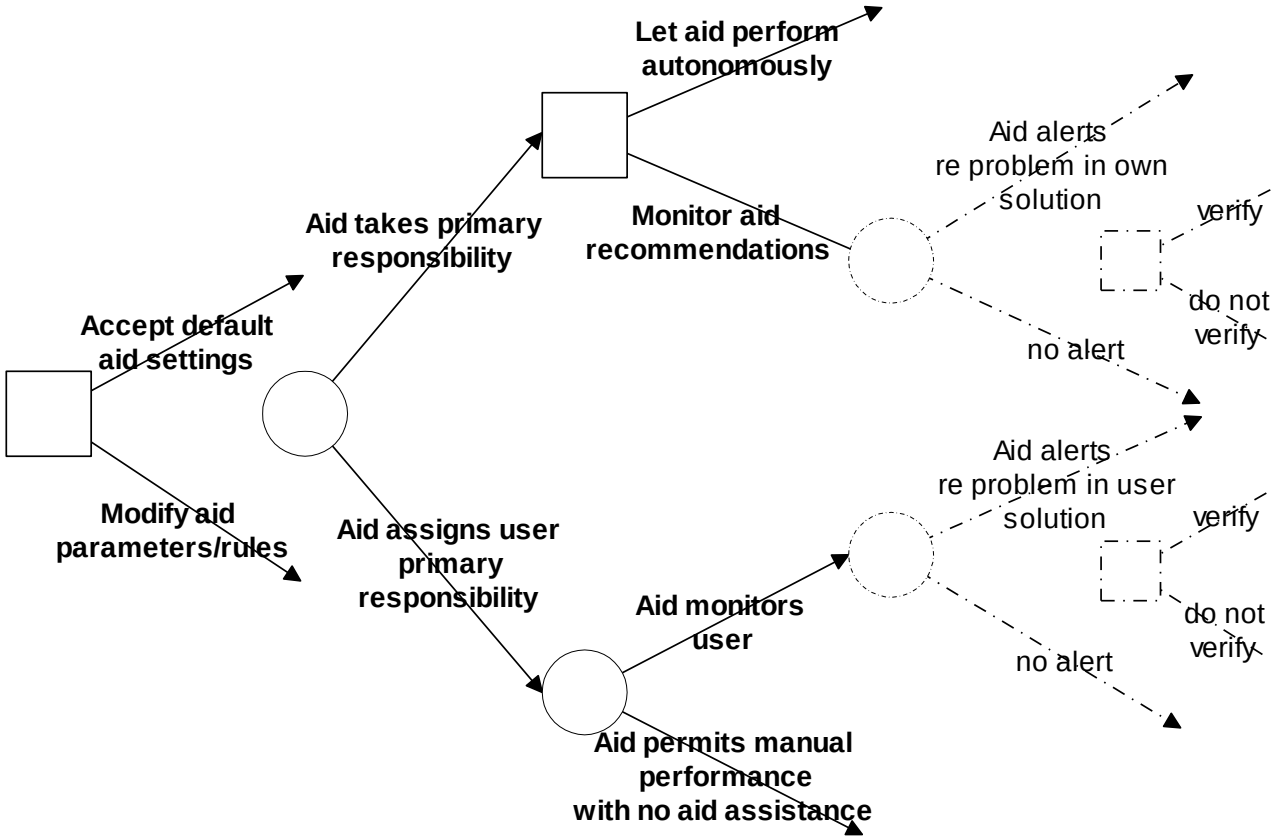

Figure 20. Illustrative Phase 2 design for an adaptive aid, with circular nodes controlled by the aid and square nodes by the user.

Figure 19 and Figure 20 are only two points in a space that contains many other possible combinations of adaptability and adaptiveness in aid design. (Both adaptability and adaptiveness, of course, presuppose that the aid's modes of operations are flexible. Thus, another dimension of this space is the rigidity versus flexibility of the aid's mode of operation.) Any combination of aid and user control over the three broad elements we have identified (primary responsibility, secondary responsibility, and adjustment of parameters and rules) is possible, with respect to almost any task performed by the aid or user. The choice of a design from this space is a Phase 1 design decision. As indicated in Figure 21, this decision is also subject to representation in a decision tree format. The outcomes of Phase 1 design choices are represented by the differences in the Phase 2 and Phase 3 decisions trees that they produce – i.e., which Phase 2 and Phase 3 events are treated as user decisions and which, from the user's point of view, as uncertain events.

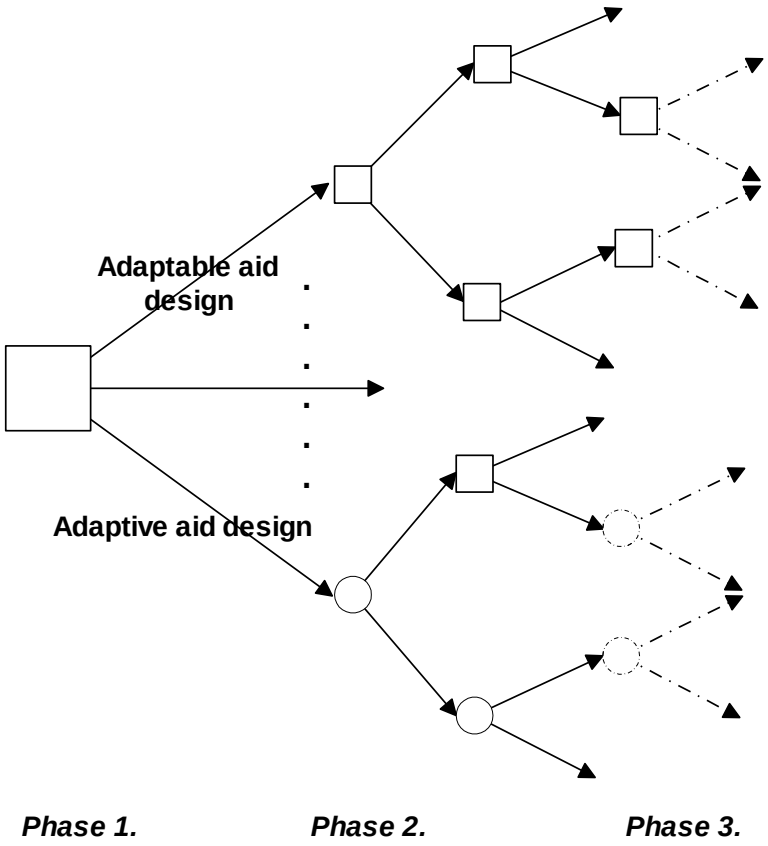

Figure 21. Phase 1 design decisions determine the degree of adaptability vs. adaptiveness of the aid in Phase 2, which in turn influences possible user actions in specific Phase 3 tasks.

Decision Trees for Supervisory Decisions

The higher-level supervisory decisions in Phase 2 involve a larger temporal scope than the more specific supervisory decisions of Phase 3. For example, a single automation mode decision in Phase 2 may influence the user's response to multiple aid recommendations in Phase 3. The distinction between Phase 2 and Phase 3, however, can sometimes be blurred. For example, a flexible aid will permit a user to shift automation modes (a Phase 2 "planning" decision) as frequently as the user likes, possibly even as often as the aid arrives at new conclusions in Phase 3. However, automation mode decision making is far from cost-free. Even though commercial airline pilots can change the automation mode of the Flight Management Computer at any time, Sarter & Woods (1992) emphasize the cognitive workload demanded by keeping track of and evaluating the various modes that can be selected. Because of these costs, users may not choose to revisit the automation mode question as often as they could, and even in flexible systems automation mode decisions will usually span a temporal scope greater than a single aid conclusion. Temporal scope, in turn, implies a need to predict the factors that will influence performance by the aid and by the user during the period until the next automation mode decision. Because it is earlier in the decision tree, the automation mode decision is typically made with less information than the verification decision. At the least, the user does not know what recommendations the aid will actually make in the future, or what confidence it will report. The automation mode decision may be thought of as based, implicitly, on probabilities across a large number of unknown events. *The user must choose an automation mode based on a higher ratio of prediction to known fact, and without knowledge of the specific aid conclusions to which it will apply*.

Figure 22 illustrates Phase 2 supervisory decisions regarding primary and secondary responsibility for task, made by users of an adaptable aid like the one shown in Figure 19. First, users determine whether they or the aid will have primary responsibility for a task. If they assign primary responsibility to the aid, they must decide whether to monitor or not. If they assign primary responsibility to themselves, they must decide whether to use the aid to verify their own conclusions.

These supervisory decisions are based on evaluations of cost in comparison to different kinds of trust, i.e., expectations regarding successful future system performance from the vantage point of Phase 2. For example, the

user's decision whether or not to monitor the aid depends on a the cost of monitoring and on the difference between trust in the aid by itself (i.e., successful performance in an autonomous mode of operation) and trust in user-aid collaboration due to monitoring. If the expected improvement in trust caused by monitoring exceeds its cost, the user should monitor. This is precisely the kind of decision that is illuminated by value of information methods.

Value of Information for Supervisory Decisions

Users who choose to monitor the aid will collect several kinds of information that users who choose automatic performance will not:

1. First, they will make general observations pertaining to the reliability of the aid. For example, in Figure 22 the user might observe that since the mission is in desert terrain, rotorwash is likely. (Rotorwash, of course, is observed later, so it is to the right of the part of the tree shown in Figure 22.) These observations do not require the user to know what recommendations the aid will actually make. They are relatively low cost in comparison to the insight they might provide into future aid performance.
2. Second, users who decide to monitor will look at specific aid recommendations, including the type of aid decision (e.g., is this a recommendation regarding battle position or a recommendation regarding attack route?), the specific content of the recommendation, and, possibly, the aid's level of confidence. Figure 22 illustrates one kind of information that the user may obtain by monitoring aid conclusions: the aid's reported trust in its own conclusion. Reported confidence may directly influence the verification decision, but since it is not known in advance, the monitoring decision must be based on a probability-weighted average of different possibilities. In this example, the aid has a good track record, and this user believes that it will report high confidence 90% of the time under present conditions, and low confidence 10% of the time. Reviewing each aid recommendation is not cost-free. However, the aid itself can relieve some of the workload by calling the user's attention to potential areas of difficulty, as illustrated in the adaptive aid diagram in Figure 20. (See Cohen, Thompson, & Freeman, 1997, for a successful experimental test of this concept.)
3. Thirdly, users who monitor may decide to collect even more information by verifying the aid conclusion. Collecting this third type of information is not part of the original monitoring decision. *Monitoring is a commitment to make the verification decision*, as Figure 22 indicates, based on general information and knowledge of the specific aid recommendation, but it does not guarantee that the outcome will be a decision to verify. Verification, of course, incurs still higher costs in exchange for more detailed information about a specific aid recommendation.

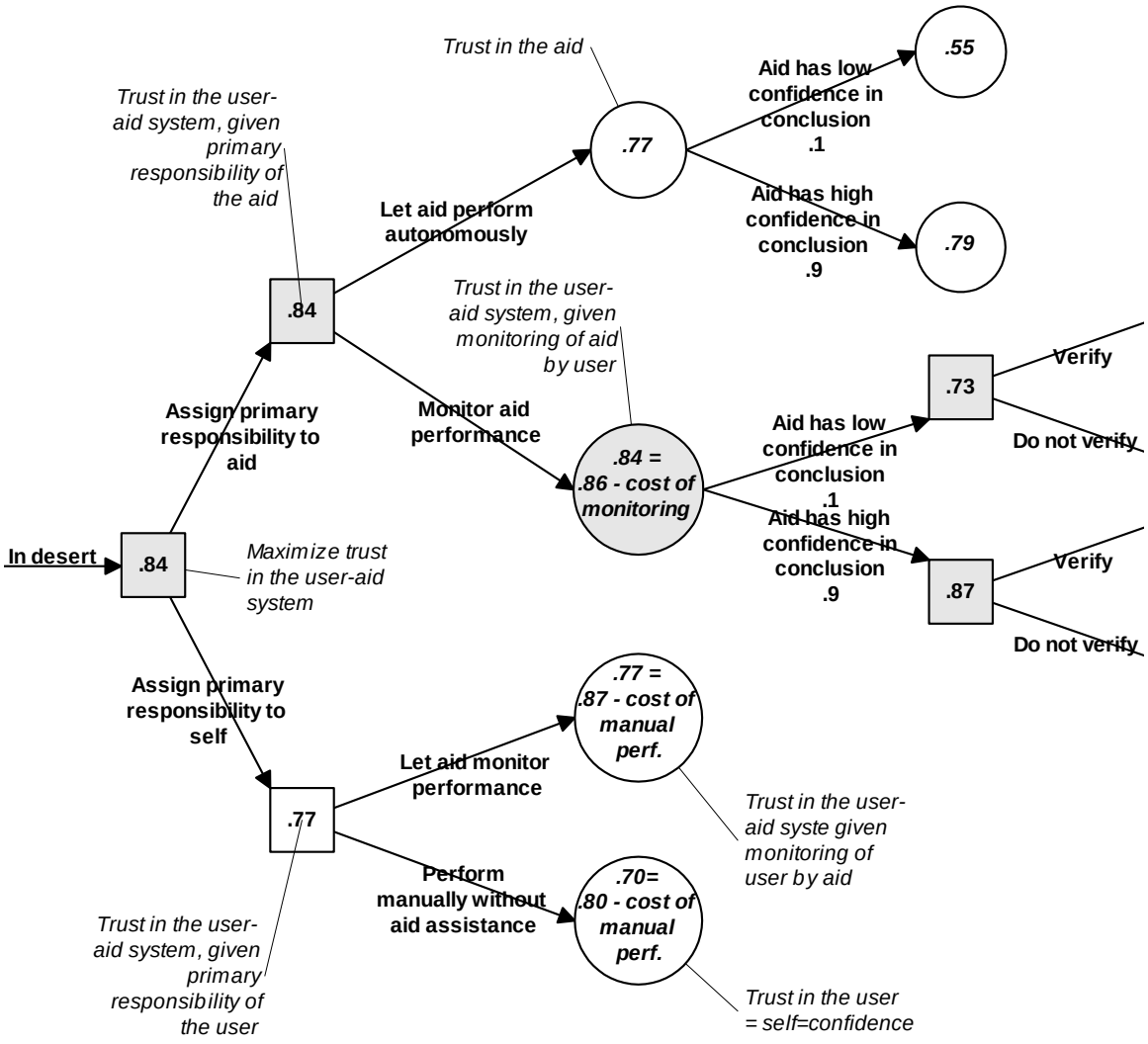

Figure 22. Example of Phase 2 automation mode decisions. Shading indicates the sequence of nodes the user may traverse, based on maximizing utility and on chance.

Monitoring has value to the extent that it can ultimately change the user's mind about specific aid recommendations. Monitoring can do this in two different ways: without verification and with verification. First, general information about the context (e.g., there are few if any enemy vehicles in a specific area) and knowledge of the type and/or content of the aid conclusion (e.g., identification of a ground contact as an enemy), perhaps combined with the aid's own low confidence, may cause a user to reject an aid recommendation out of hand (e.g., conclude immediately that the vehicle is probably friendly). Secondly, monitoring can cause a change of mind by the user indirectly, by leading to further verification of the aid conclusion (e.g., looking more closely at the image of the ambiguous contact).

The concept of value of information applies to Phase 2 supervisory decisions just as it did to the verification decision. This measure balances the advantage of gathering more information against the costs of gathering the information. A verbal formula for value of information, applied to the monitoring decision, is the following:

Value of monitoring = expected number of decisions in the time covered
** Sum over all observational outcomes relevant to monitoring:*
[probability of the observational outcome
** expected change in trust per decision due to the observation*
- expected cost per decision of making the observation]

The user should monitor the aid's recommendations if this value if greater than zero.

As a result of the two different ways that monitoring can cause change of mind, the "benefit" component of this formula breaks down into two parts, corresponding to the Phase 3 user decision to verify or not to verify a particular aid conclusion:

expected change in trust per decision due to the observation =
*prob (verification) * expected change in trust per decision with verification*
*prob (no verification) * expected change in trust per decision without verification*

Of course, the expected change in trust due to verification incorporates both the cost and the expected change in utility due to verification, as discussed earlier in this chapter. (In fact, the node in Figure 20 representing the user's decision to verify, with 87% trust in a successful outcome, is the starting node in the verification decision tree of Figure 14.)

Value of information captures the tradeoffs in the monitoring decision. If users choose an autonomous mode of operation for the decision aid, virtually no user effort in Phase 3 is demanded. But since users have chosen not to monitor the aid, they are committed to accepting aid conclusions in both high and low confidence conditions. As can be seen in Figure 22, this represents a significant swing in accuracy, from 79% when confidence is high to 55% when confidence is low. By contrast, when users monitor the aid, this swing is reduced to 87% when confidence is high and 73% when confidence is low. Users will be more likely to verify an aid conclusion, and improve performance, when the aid declares itself uncertain. The penalty for monitoring the aid's conclusions in this example is quite low, only about 2% chance of success. Yet it improves the chance of success by 10% over automated performance.

Thus far, we have focused on the case in which users allocates primary responsibility in a task to the aid. When users takes primary responsibility for themselves, they face a similar choice, whether or not to utilize the decision aid as a source of verification for their own conclusions. This decision lends itself to a very similar value of information analysis. The extra observations that the user considers making involve the aid's conclusions, its level of confidence, and its explanation of its reasoning. Making these observations has a cost in time, but it may draw the user's attention to factors overlooked by users and, on some occasions, cause them to change their response. Once again, Figure 22 illustrates the advantage of combining the resources of the user and the aid. When the aid is utilized to verify user conclusions there is about a 7% net increase in chance of success.

Another supervisory decision in Phase 2 is the decision regarding primary responsibility. Leaving costs aside, in this example users on their own happen to do better (80% appropriate) than the aid on its own (77% appropriate) in selecting battle positions. However, the manual mode (i.e., user with primary responsibility for the task) is labor intensive, costing users a 10% reduction in chance of success due to delay and distraction from other tasks. (This might be the case, for example, if users have paper maps with more detailed vegetation and cultural features than a Combat Battle Position Recommendation aid.) Monitoring is a way of resolving this tradeoff between autonomous performance and manual performance, providing many benefits of both at only a fraction of the cost of manual performance. In Figure 22, as in many real situations, the user and the aid have complementary capabilities. Because it combines the knowledge of human and machine, the monitoring strategy may do better than either the human or the machine alone. Because it leaves most of the work to the machine, and involves the human only when needed, it may be far more efficient.

Finally, in Phase 2 users may have the opportunity to adjust some of the parameters or rules that the aid will use when it generates Phase 3 recommendations. Adjustment of parameters or rules is the Phase 2 equivalent of modifying aid recommendations in Phase 3: For example, suppose a user does not trust a target identification aid's ability to discriminate tanks from armored personnel carriers (APC's). The user may wish to replace classifications by the aid as *tank* or *APC* with more general classifications reflecting the user's lack of confidence, such as *tracked vehicle*. On the other hand, suppose a user knows from intelligence sources that there are no APC's in the area. This user might wish to replace any aid classification of a vehicle as *APC* or *tracked vehicle* with the specific classification *tank*. These results may be accomplished ahead of time if the aid permits excluding certain possible classifications (or weighting cues so that those classifications are less likely.) Adjustment of parameters is simply the adoption of a consistent *policy*, corresponding to a proactive and systematic choice of Phase 3 reactions to aid recommendations. The advantage of adjusting aid parameters ahead of time is not only consistency, but economy of effort. In addition, the work done by the user is transferred from Phase 2, where time is typically critical, to Phase 3, where time is less critical.

The adjustment decision in Phase 2 is subject to the same value of information analysis that we applied to other user interactions with the aid. Part of the value of adjusting parameters, in fact, may be to make it possible to avoid subsequent interactions (such as verifying aid recommendations). Recall that in the example of Figure 14. the user of a Combat Battle Position Recommendation aid decides to verify a recommended battle position based on a 25% chance of an unacceptable angle of attack in desert terrain. The expected benefit of verifying angle of attack was a

12% improvement in chance of success, from 77% to 89%, with a 2% cost due to time, to 87%. The benefits of verification can be obtained with less cost, however, if the aid allows the user to exclude specified regions ahead of time. Suppose that the user has determined in Phase 2 which parts of the area were likely to be used by the enemy as avenues of attack, and designates these areas as *exclusion zones.* As a result, the user's trust in the aid by itself coming up with an acceptable battle position rises from 77% to 93%. Verifying specific battle positions for angle of attack has become unnecessary.

Constraints on the Automation Mode Decision

In our discussion of the verification decision, we saw that it is often impossible to specify in advance the kinds of information that users might obtain when verifying an aid recommendation. For example, perceptual inspection of an image identified by a target recognition aid, or critical thinking about conflicting goals in the selection of a battle position, may lead to unanticipated observations or insights. This uncertainty affects the evaluation of automation mode options as well, for the same reasons. The value of monitoring might be underestimated if we only consider features of the situation or even of the aid conclusion that can be explicitly identified in advance. Fortunately, this problem can be addressed by the same method we used for verification decisions: by deriving constraints on the appropriateness of monitoring. If such constraints are not satisfied, monitoring cannot be worthwhile, no matter how much additional information is obtained.

Three automation mode options may be ordered in a sequence of increasing costs in exchange for increasing benefits: automated performance, in which only information available to the aid is utilized in a decision; primary responsibility by the aid, with the user monitoring, in which information available to the aid is supplemented by the decision-aid verification skill of the user; and primary user responsibility, with vetting of user responses by means of the aid (in some but not all cases, the latter produces the best performance but at the highest cost). The choice among these strategies depends on trust in the aid, construed as the expected proportion of appropriate decisions during the time period covered by the supervisory decision. We can derive upper bounds on trust for the appropriateness of each of the three supervisory strategies (just as we did for the three verification strategies in Figure 18). Again, just as in Figure 18, we assume that the effects of benefits do not increase as rapidly as the effects of costs as we "upgrade" to more costly and more beneficial strategies. Thus, the more effortful strategies require a higher degree of distrust in the aid before they will be adopted.

The horizontal axis of Figure 23 represents the average time stress (or expected cost of delay per decision) during the period covered by the supervisory decision. For example, a user performing ground-based planning the day before a mission will be at the extreme left, representing low time stress. A user who is replanning in the cockpit will be on the extreme right of the figure, representing high time stress. Even if they have the same level of trust in the aid, the first user may select a more manual mode of performance, while the second opts for a more automated mode of operation.

The upper bound on trust is also affected by the expected *stakes*, or costs of errors during the period in question. This is the difference that a good decision versus a bad decision will make, on the average during the relevant period, in terms of mission success. For example, cost is low if there are many good battle positions, so that selecting the very best ones does not matter much. Cost is high if there are very few adequate battle positions, so selecting non-optimal ones could mean failure of the mission.

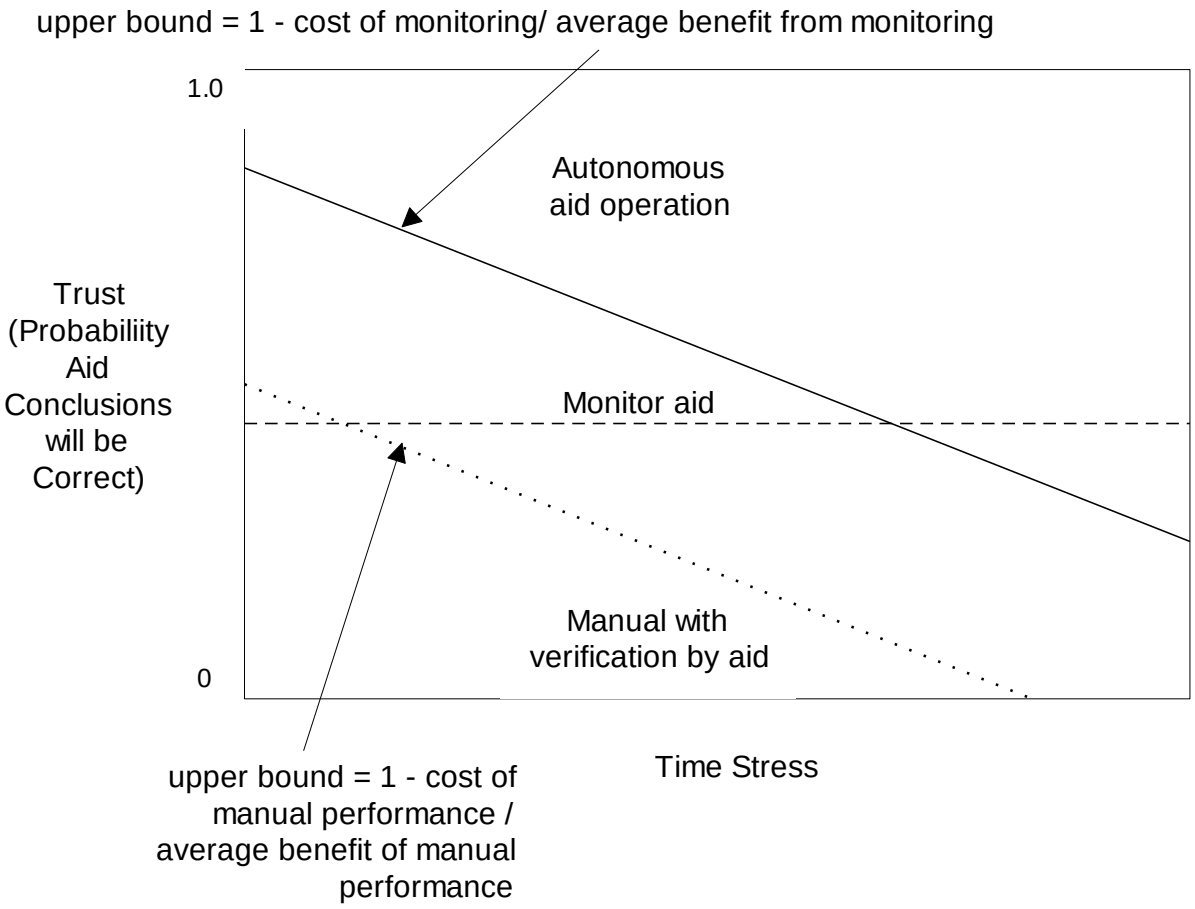

Figure 23. Benchmark model for selecting an automation mode. The horizontal long-dotted line is an illustrative level of trust in the aid.

The most important difference between this figure and Figure 18 is that the parameters of the model are aggregated measures over the relevant period of time, rather than probabilities and utilities of specific events. We saw in Chapter 2 that trust in this Phase 2 context can be interpreted as the expected relative frequency of correct system responses, rather than the probability that a specific conclusion is correct (Phase 3). Trust in the aid is, in effect, the probability that autonomous operation will be the correct choice of automation mode across the set of expected decisions. This is equivalent to the user's average trust in the option of accepting every specific aid recommendation without verification. Similarly, costs of errors reflect the costs of incorrectly acting on aid recommendations, averaged across the types of recommendations the aid is likely to make in the relevant period of time. Finally, time stress reflects the average cost of the time that a particular strategy is expected to require; it varies both with the risk and cost of time, and with the actual amount of time a strategy is likely to consume. We saw in our discussion of Phase 3, that more knowledgeable users are likely to take less time performing tasks like verifying aid conclusions (because their trust judgments are already likely to be more highly resolved). This efficiency reduces expected time demands for interacting with the aid on the part of the very individuals who have the most to contribute. Thus, another conclusion that we can now draw is that *knowledgeable users are more likely to select modes of aid operation that are not fully automated.*

Training Implications: Generation of Scenarios and Feedback

Training Content: Patterns that Cue Interaction Decisions

Reliance decisions must be made quickly and "intuitively." If they take much time, users will incur the risks of delay without any of the benefits. For example, thinking in Phase 3 about whether or not to verify an aid's conclusion can take time away from actually doing so. Or thinking in Phase 2 about whether to select an autonomous aiding mode may rob users of the time for taking advantage of automation. The premium on speed increases as the temporal scope of the decision decreases. Thus, the decision at Phase 3 about whether to verify an aid's recommendation must be made more quickly than a decision at Phase 2 about the automation mode to be utilized over a longer period.

The goal of training is not to teach users to construct models of the kind we have been examining. Benchmark models are not meant to represent the thought processes of users in any literal sense. Rather, the goal of training is to sensitize users to *patterns* of cues that can be quickly and intuitively recognized. Such patterns are the real content of training. They determine whether verifying an aid's conclusion is worthwhile in Phase 3, or whether a fully automated mode should be selected in Phase 2. Benchmark models, however, are a valuable tool for identifying the elements of the patterns to which users should be sensitive, and the manner in which they should respond to them. We saw in the previous sections that such patterns can be simply characterized in terms of uncertainty, time stress, and stakes.

Training Tools: Scenarios and Feedback

Benchmark models can be used to set up a series of training scenarios in which different reliance decisions are appropriate. Such scenarios can then be used to provide practice and feedback to decision aid users in making appropriate reliance decisions. These relatively models can serve as the starting point for training rapid, intuitive judgment.

Figure 24 through Figure 27 show how a set of training scenarios for Phase 3 decisions might be generated by systematically manipulating two of the three key variables — time stress and stakes. For this example, we have kept trust constant, at .4 chance that the aid's recommendation is correct. As in earlier sections, the aid has recommended that a contact be engaged. Stakes are varied for the upper bound only, by manipulating the mix of friendlies and enemy non-targets, thus affecting the expected cost of a mistaken engagement. Time stress is varied by manipulating the rate of increase in the danger of being targeted, as the user spends more time unmasked.

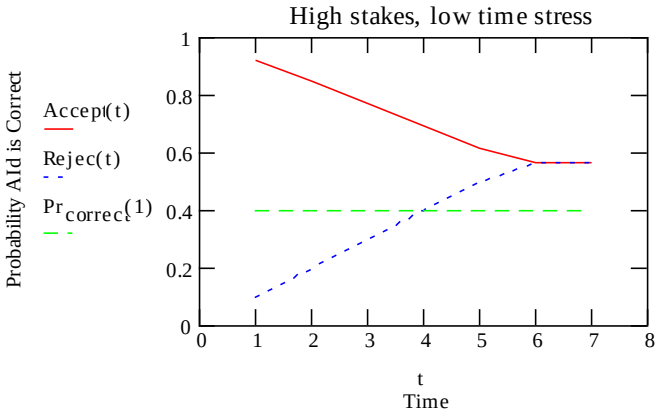

Figure 24. Scenario in which there is a large proportion of friendlies relative to enemy non-targets, producing high stakes of incorrectly accepting the aid's recommendation to engage. The probability of being targeted by enemy platforms is low, but increases with time. Trust is highly uncertain, at .4. The result is a significant amount of time (from time 1 to time 4) spent verifying the aid's recommendation to engage. Finally, the cost of remaining unmasked leads to a decision (in this case, not to engage).

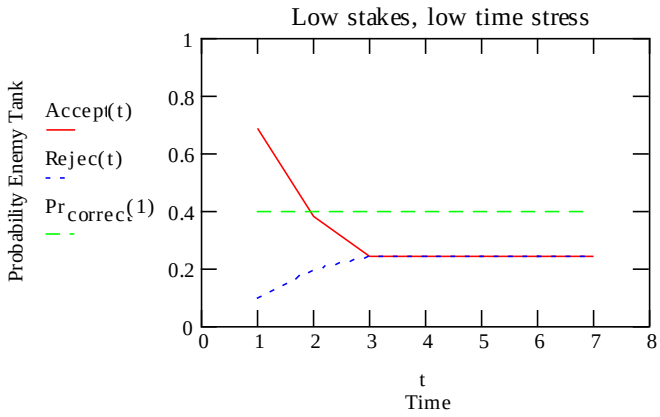

Figure 25. Scenario in which the low proportion of friendlies relative to enemy non-targets leads to a low threshold for engagement. Even though time stress is low (as in the previous example), less time is spent verifying the aid's recommendation (from time 1 to time 2) because of the low cost of an error. A relatively quick decision is made to engage.

In these scenarios, the user (or trainee) must decide not only what to do — i.e., whether to engage or not to engage a contact — but how long to wait before doing it. In two of the scenarios (Figure 25 and Figure 27), the appropriate action is to accept the aid's recommendation and engage, while in the other two (Figure 24 and Figure 26), the appropriate action is to reject the aid's recommendation and not to engage. The appropriate time spent verifying the aid's recommendation varies from 3 units (in Figure 24) to 1 unit (in Figure 25 and Figure 26) to 0 units (in Figure 27). Trainees can be evaluated and given feedback on both of these dimensions. Exercises of this kind can help maintain skills in the primary task, while enhancing the ability to interact effectively with a decision aid.

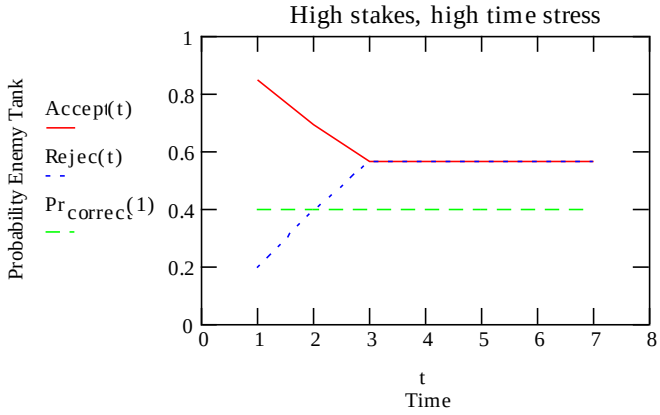

Figure 26. Scenario in which the cost of a mistaken engagement is high, due to a high proportion of friendlies. However, time stress is also high, due to a rapid increase in the chance of being targeted with time spent unmasked. This results in a relatively early decision, in this case not to engage.

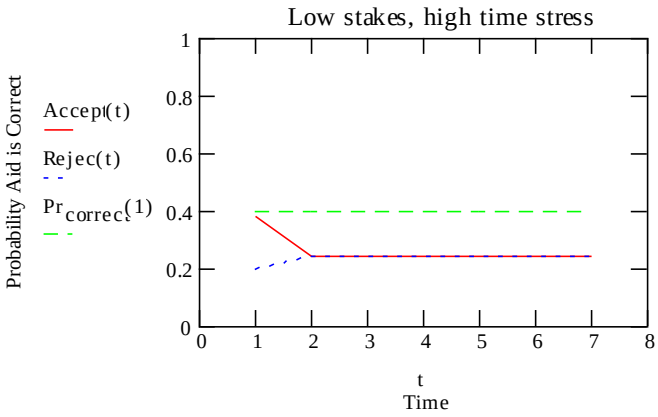

Figure 27. Scenario in which the cost of a mistaken engagement is low (due to low proportion of friendlies) and time stress is high (due to rapidly increasing chance of being targeted). The result is no time spent verifying aid's recommendation, and an immediate decision to accept the recommendation to engage.

Training of this kind might progress to patterns of increasing sophistication. In the above examples, the upper and lower bounds were independent of trust in the aid's conclusion, and trust remained constant. As Figure 28 illustrates, however, neither of these conditions is necessary. In this example, trust begins, as before, at .4. However, in verifying the aid's recommendation to engage, the user finds evidence that supports the aid's identification of the contact as hostile. Thus, confidence in the recommendation to engage increases to above .8. As the user becomes increasingly convinced that the contact is hostile, there is also a rise in the perceived chance of being targeted. In short, time stress increases along with trust. The result is a somewhat earlier decision to engage the target, as compared with Figure 24.

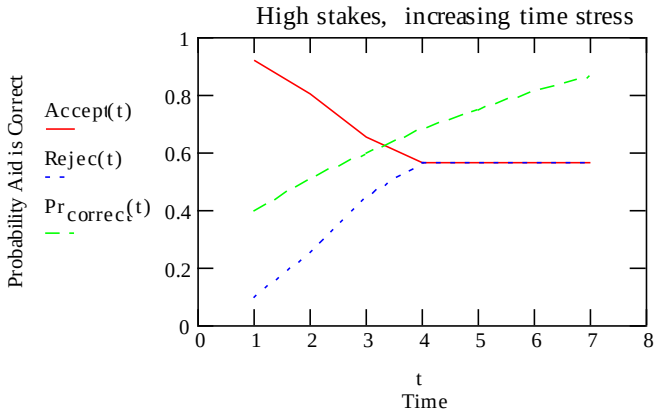

Figure 28. Scenario in which trust in the aid's identification of the contact as hostile increases, bringing with it an increase in time stress due to the expectation of being targeted. The result is a somewhat earlier decision to engage than in Figure 24, which is otherwise based on the same underlying parameters.

A FRAMEWORK FOR DECISION AID USER TRAINING REQUIREMENTS

The benchmark models in Chapter 3 and Appendix A addressed two kinds of reliance decisions: whether or not to verify a particular decision aid conclusion, and the selection of an automation mode, respectively. In this chapter, we attempt to provide a more general overview of user decisions with regard to decision aid use. We will identify a variety of user supervisory tasks, such as setting automation mode (as previously discussed), monitoring system performance, diagnosing problems in system performance, and taking appropriate corrective action. Each of these supervisory tasks can in turn be applied to a variety of decision aiding functions (such as data aggregation, situation assessment, option generation, and so on). And each combination of supervisory function and decision aid function may be associated with potential pitfalls or problems, which help to generate training requirements.

Event trees and benchmark models, such as those explored in the previous chapters, can be developed for each of the user supervisory tasks. An important factor in evaluating these models is the extent to which they address all or most of the problems associated with each combination of supervisory task and decision aid function. The event trees and benchmark models, in turn, help refine the training requirements for decision aid users: The goal of training is to bring user performance into as close a correspondence as possible to the knowledge and behavior specified in these models.

Potential Problems in User Interaction with Decision Aids

One starting point for developing a more general overview of training requirements for decision aid use is to identify the potential problems than can arise in user-aid interaction. Table 8 shows an extended though not comprehensive listing of potential problems. This can be thought of as a generic list, irrespective of specific decision aiding functions.

As an example, consider P16, Ignoring Alerts Due to False Alarms. Users may ignore automation or decision aids because of frequent false alarms — alerts that turned out not to be emergencies, or advice that was not appropriate. As a result, users may under-utilize a decision aid, even though its use might be critical in certain conditions — the classic case of the cry-wolf syndrome. Training could counter this syndrome. P13, Decision Biases, refer to consistent patterns of human decision making style that do not confirm to prescriptive or normative models (Cohen, 1993; Tversky & Kahneman, 1974). For example, many users display a confirmation bias, or a tendency to seek evidence that verifies a "pet" hypothesis rather than data that could potentially disconfirm it. Such a bias could hinder effective monitoring of a decision aid by the user so as to check for potential problems in aid performance. Several other potential problem areas are listed. Some caveats and uncertainties should be noted at the outset: Not all of these potential difficulties will apply to each specific aid or aiding function; the relative importance of these problem areas will vary with aiding functions; and interactions between factors (e.g., high workload and poor understanding of action risks) are not well understood (Riley, 1994). Nevertheless, there is good evidence for the influence of each of these factors in at least one empirical study (see Appendix A).

Training Requirements

The research data base in Appendix A, as well as other studies, can be examined to identify training procedures that could be used to rectify the problems areas described. The training techniques can be divided into:

- training for human-automation interaction, in general
- training for human interaction with decision aids, in general
- training for specific human operator functions with specific decision aiding functions

Training for human-automation interaction in general is too big a topic to be reviewed here. Instead, we focus on training for decision aid usage, both in general, and with reference to specific functions.

Decision aid functions may be categorized in a number of ways, including by levels of automation, as described earlier. Sheridan (1992), for example, listed 10 levels of automation. For the purpose of identifying training requirements for decision aids, however, it may be sufficient to divide these functions into five broad categories, *data aggregation, situation assessment, option generation, action selection, and action* (see Table 9). These categories tend to correspond to increasing levels of automation and complexity, from simple data aggregation and integration, to action selection and implementation. However, each of these functions can involve different degrees and types of user involvement.

Table 8. Potential problems in human interaction with decision aids

P1. High Overall Workload
P2. Fatigue
P3. Low Overall Workload (Boredom)
P4. High Cognitive Overhead of Engaging Aid
P5. Low Self Confidence in Manual Skills
P6. Inappropriately High Self Confidence in Manual Skills
P7. Poor Understanding of Action Risks
P8. Limited Understanding of Aid Functions
P9. Incomplete Mental Model of Aid
P10. Complacency
P11. Monitoring Paradox
P12. Mode Confusion
P13. Decision Biases
P14. Inappropriate Use of Decision Heuristics
P15. Difficulty in Alternative Option Generation
P16. Ignoring Alerts due to False Alarms
P17. Poor Workload Management
P18. Cognitive Tunnel Vision
P19. Poor Time Management
P20. Team Communication Barriers

Training requirements for each of these modes of decision aid functionality are typically specified in most instructional programs for decision aid use. For example, the user may be taught what inputs are required for data aggregation, how to recognize the outputs that are displayed, and how to use these outputs for action selection and implementation. As noted previously, however, this training falls short of what users need for the appropriate use of decision aids.

Identifying these training requirements can proceed by distinguishing between different user *supervisory functions* with respect to the use of the aid. Each decision aiding function is subject to different modes of user involvement by virtue of these supervisory functions. Eight such user functions are listed in Table 9. This is not meant to be a comprehensive list, but a starting point for analysis. Anchoring the list at each end are the functions of engaging and disengaging the aid, or a particular decision aid functionality. On the surface, these might appear to be simple tasks that do not require much training. As Riley (1994) has indicated however, the decision to engage or to disengage an automated function may be one of the most important a human user can make, particularly in time-critical situations. Parasuraman and Riley (1997) describe several transportation accidents and incidents that were associated with the human user's decision to use or not use an automated function. In between the functions of engagement and disengagement are several other user supervisory functions: *selecting automation mode, assessing aid performance, monitoring problems in aid, diagnosing problems in aid, action selection, and action.* We explored two of these functions, selecting automation mode and monitoring aid problems, in the previous chapter.

Table 9 displays decision aid and user supervisory functions in a matrix form. Training requirements may vary with each combination of decision aid and user function. Our initial analysis suggests that some combinations do present particular challenges to training , and we have indicated these by bullets. These key functions may also be the ones that when fully trained contribute most to system efficiency. Two aid functions are discussed more fully to illustrate these training needs.

First, data aggregation functions should typically pose significant challenges to user training only for the functions of monitoring and diagnosis. For example, training to engage, disengage, or select aid mode should be relatively straightforward when only data aggregation is involved. Training the user to monitor and diagnose this aid function, on the other hand, may be more difficult. In terms of the trust model described above (Chapter 0), there are two difficulties: First, *trust* in this function is typically high. By definition, data aggregation represents a fairly low level of automation. This level of automation is also likely to be relatively reliable, given that it will typically involve fusion of multiple sensor data and limited, if any application of computer "intelligence." Nevertheless, data aggregation may not be perfectly reliable, sensors may fail etc. As discussed previously, monitoring of high-reliability systems is extremely difficult for users. Human operators typically do not allocate attentional resources to monitoring of systems that fail only occasionally if they are loaded with other tasks. This "complacency" effect has been noted for a wide variety of systems (Parasuraman et al., 1993). It corresponds to problem P10 in Table 8.

Potential problems in human interaction with decision aids. Moreover, a tendency to accept aid conclusions without monitoring is predicted by the benchmark model for selection of automation mode presented in the previous chapter. The second problem concerns the inability to observe features that cue data aggregation difficulties. Users may also not have access to all the "raw" data prior to data aggregation, because it takes place at such a basic level, so that *diagnosing* problems in this function may be difficult even if the user does monitor the automation and detect an error. This corresponds to problem P9 in Table 8. Potential problems in human interaction with decision aids, an incomplete mental model of the aid. In terms of the APT model (Figure 2), it reflects incomplete knowledge about and/or incomplete awareness of features that are associated with poor aid performance.

Table 9. Framework for developing training requirements for human interaction with decision aids. Bullets indicate functions with the greatest training need and that may profit most from successful training.

		Decision Aid Functions				
User Functions		M1 Data Aggregation	M2 Situation Assessment	M3 Option Generation	M4 Action Selection	M5 Action
H1	Engaging Aid				✓	✓
H2	Selecting Automation Mode		✓	✓		
H3	Assessing Aid Performance		✓	✓	✓	✓
H4	Monitoring Problems in Aid	✓	✓	✓		
H5	Diagnosing Problems in Aid	✓	✓	✓		
H6	Action Selection		✓	✓		
H7	Action		✓			
H8	Disengaging Aid				✓	✓

Parasuraman et al. (1996) showed that adaptive function allocation, or occasional manual performance of an automated function, had a carry-over effect on subsequent monitoring of the automation. This beneficial effect was attributed to the refreshing of the user's memory for the inputs and operations of the automated task (i.e., leading to an more complete mental model of the situation and the task). However, this may not be a practical solution for improving monitoring of data aggregation functions because there are so many of these functions in any given system. Parasuraman et al. (1996) recommended adaptive function allocation primarily for high-level automation functions. Improving monitoring and diagnosis of automated data aggregation may therefore require one of two approaches: (1) no specific training, but display design changes that allow data aggregation anomalies to be detected more easily (configural displays, etc.); (2) training to increase awareness of the possibility of data aggregation errors, of features might cue such errors, and of the consequences of poor monitoring.

In contrast to data aggregation, the *situation assessment* function presents a number of challenges to training across a wide range of user supervisory functions. This function is critical to effective use of the decision aid in uncertain environments. Engaging or disengaging the aid need not require unusually difficult training; users could be told simply to engage the aid whenever possible. However, training will be required to assess the aid's performance, as well as other functions (see Table 9). This might take the form of (1) a better understanding of strengths and weaknesses of the algorithms used by this aiding function, (2) improved calibration of trust assessments as a function of relevant features that affect the reliability of situation assessment, and (3) more efficient strategies for allocating attention. With respect to the latter, training might help users learn not to spend significant amounts of cognitive resources in diagnosis and assessment of the aid in highly time-critical situations. Such training might include scenarios and feedback shaped by benchmark models of the kind discussed in the previous chapter.

Training Requirements for User-Decision Aid Interaction

We have identified five training requirements for enhancing user interaction with decision aids:

1. Conveying more complete mental models, which include key features of the system, domain, situation, task, and aid conclusion that are related to the quality of system performance.
2. Development of critical thinking skills for handling novel or highly uncertain situations.
3. Development of situation awareness, or monitoring skills, for identifying situations where mental models or critical thinking skills should be applied.

4. Ability to generate trust assessments that accurately discriminate among situations where aid performance is significantly different.
5. Learning appropriate strategies for interacting with decision aids at different phases, and learning the conditions under which each strategy is appropriate, in terms of trust, workload, stakes, and confidence in manual performance.

These five requirements, taken together, encompass the crucial elements and relationships of the APT-R model, as we saw in Figure 13.

Perhaps a more important question is this: How successful will these three strategies be in addressing the problem areas associated with human interaction with decision aids? Table 10 makes a start at answering that question. It organizes the 20 areas that we identified in Table 8 according to the training strategy that seems appropriate. It suggests that these three strategies not only cover all the basic concepts surrounding user-decision aid interaction (as represented in APT-R), but have the potential for addressing all 20 problems as well.

Table 10 permits the identification of appropriate training methods to target specifically identified problems. For example, the impact of decision biases could be reduced by training in all five content areas: by showing users features and situations when an aid function is likely to yield suboptimal performance; by part-task training in assessing the aid's ability to deal with different kinds of problems and the "calibration" of trust in the decision aid; and by practice in strategies that maximize probabilities of successful performance. To be maximally useful, these training methods would have to be tailored specifically to the matrix of aid and user functions shown in Table 9.

In the remainder of this chapter, we pursue this point further with a somewhat more detailed discussion of decision biases.

Table 10. Mapping of problem areas in user-decision aid interaction to the training strategies that might address them.

Training Content	**Problem Area**
1. Mental Models of features that determine aid strengths and weaknesses 2. Situation Awareness regarding such features 3. Critical thinking to detect problems	P8. Limited understanding of aid functions P9. Incomplete mental model of aid P12. Mode confusion
4. . Calibration of trust and self-confidence 3. Critical thinking to reduce over-confidence	P5. Low self-confidence in manual skills P6. Inappropriately high self-confidence in manual skills P7. Poor understanding of action risks
5. Strategies for reliance decisions 2. Situation awareness skills for monitoring efficiency	P1. High overall workload P2. Fatigue P3. Low overall workload (boredom) P4. High cognitive overhead of engaging aid P17. Poor workload management P18. Cognitive tunnel vision P19. Poor time management P10. Complacency P11. Monitoring paradox P20. Team communication barriers
All 5 training requirements (see next section)	P13. Decision biases P14. Inappropriate use of decision heuristics P15. Difficulty in alternative option generation P16. Ignoring alerts due to false alarms

Decision Biases

The user's interaction with a decision aid is itself a decision process. It involves the integration of information to arrive at judgments of trust and confidence (e.g., predictions of the probability of correct actions by the aid and by the user); once such assessments are made, it involves decisions regarding the allocation of functions between the user and the aid. A major finding of cognitive science research on decision making (e.g., Kahneman, Slovic, and Tversky, 1982) is that unaided decision processes employ simplifying heuristics that at best only approximate prescriptively accepted rules (e.g., Bayesian decision theory). Cohen (1993) presents a naturalistic interpretation of these decision errors. Such biases may affect almost every stage of a human's interaction with decision aids, and must be attended to in the design of any training for decision aid users:

Direct Assessment of Uncertainty / Trust

A number of studies show that people consistently overestimate their degree of certainty regarding predicted events and estimated quantities, even in areas where they are (rightfully) regarded as experts (Kadane and Lichtenstein, l982). The bias is a general miscalibration: people assign higher probabilities to expected events than the actual frequencies warrant. Lee & Moray (1994) have suggested that this bias may cause operators to prefer manual over automatic control even without associated performance advantages. Whether this is the case, however, may depend on the temporal scope of the assessment. When the scope is narrow, the user may have solved the problem in parallel with the aid; in that case, if the two solutions disagree, he might overestimate the probability of his own solution's being correct and override the aid. However, when predictions of aid and user performance are being made before the fact, e.g., to establish an automation mode over a period of time, overconfidence effects can apply to either probability. If an operator believes that the machine will perform well, he will overestimate the probability of a correct machine action, and be less prone to revert to manual control. In other words, the overconfidence bias should lead to inertia, i.e., a tendency to remain in the status quo.

People rely on ease of recall or imagination in estimating the frequencies of events, but these psychological characteristics of the event often do not reflect its true frequency (Tversky and Kahneman, 1973). The availability heuristic could affect predictions of aid or user performance, if the user's predictions are unduly influenced by particularly recent or salient events. A recent bad experience with a decision aid, for example, could outweigh a less memorable series of good experiences, especially in assessments that should have wider scope.

Research suggests a number of interventions or factors that mitigate biases in probability assessment (reviewed by Lichtenstein et al., 1982). Elements of each of these are captured in the training requirements based on APT-R, as indicated in Table 11.

Collecting Information / Monitoring

Wason (l960) claimed to show that people tend to seek data that confirms a favored hypothesis rather than data that would test it. This confirmation bias could hamper effective monitoring of a decision aid for indicators of problems. It might also hinder monitoring of the user's own performance for potential problems.

Parasuraman, Moloy, & Singh (1993) found that monitoring for automation failure was influenced by task load. When such monitoring was the only task, it was performed well. However, when two other manual tasks were added, many automation failures were not detected. Competition for attentional resources could not be the only explanation for poor monitoring, however. When manual performance was substituted for automation monitoring, even with the two other manual tasks present, performance was high. The subjects' performance does not seem to reflect a consistent set of criteria for monitoring.

This problem may be directly addressed by training in sensitivity to the factors that determine the appropriateness of different interaction strategies (e.g., when it is worthwhile to monitor and when not). But it may also be addressed by instruction in mental models of aid performance, in monitoring for the occurrence of features in such models, and in critical thinking that exposes and challenges one's own assumptions.

Table 11. Interventions shown by research to reduce assessment biases, and the corresponding element of the APT-R training framework.

APT-R Training Content	**Effective Interventions against Assessment Biases**
4. Probabilistic assessment training	Repetitive practice with a large number of cases Immediate trial-by-trial feedback Use of a proper scoring rule for motivation
1. Mental model / event tree training 2. Situation awareness training	A rich knowledge base, i.e., expertise or familiarity with the problem
3. Critical thinking training	Knowledge about the overconfidence bias itself. Playing devil's advocate, by generating reasons why one could be wrong.

Inferring Conclusions from Data / Evaluating Trust

People tend to ignore or discount evidence that contradicts a favored hypothesis (Nisbett and Ross, 1980). This version of the confirmation bias, or belief bias, would hinder users from acting on cues that suggest that the aid's (or the user's own) performance is degraded. Mosier & Skitka (1996) have referred to this as "automation bias." Similarly, people are slow to change their prior expectations regarding the frequencies of events as those frequencies change (Howell and Kerkar, 1981). Thus, cumulative judgments of the implications of evidence can depend critically on the sequence in which the data are presented, with early impressions being favored. Cohen and Hull

(1990) found an underconfidence effect caused by the sequence in which information was presented. They compared performance of Naval sonar classification analysts who are initially exposed to a small ambiguous segment of a sound spectrogram with analysts whose first exposure was to a larger and less ambiguous segment of the same spectrogram. Even after both groups had been exposed to the same data (the full spectrogram), the first group was less confident and took longer to arrive at a conclusion than the second group, possibly because of the difficulty of abandoning the early impression of highly ambiguous data.

These results suggest that initial experiences with a decision aid should be arranged to provide clear cues regarding its performance quality. This is precisely the goal of training in mental models of aid performance, in the form of event trees that show how confidence in an aid evolves as new cues are observed.

Choice / Decisions about Automation

When predictions are made about the outcomes of an action or plan, there may be effects of "wishful thinking" (e.g., exaggerated probability for the desired outcome) or pessimism (e.g., exaggerated probability for the undesired outcome) (Einhorn and Hogarth, l984). These effects are a function of the ambiguity of the probability estimates, and thus might be accentuated when assessments of trust are incomplete (tested under new conditions) and unreliable (based on a small sample of data). For example, a relatively new user, who had experienced an aid only under a few conditions, might exaggerate the possible negative consequences of an aid error in an unfamiliar situation. Of course, the user might also adopt a "best case" approach.

This problem is addressed most directly by training in appropriate strategies for user-decision aid interaction, and the cues that signal which strategies are appropriate. It may also be addressed, however, by critical thinking training that helps users become more aware of assumptions they may have implicitly adopted, and learn to monitor for signs that such assumptions may be getting them into trouble.

APPLICATION OF THE TRAINING STRATEGIES TO RPA

In this chapter, we describe the development of training concepts and methods guided by the framework presented in the previous chapters. The application is in the context of the Rotorcraft Pilot's Associate (RPA) program. It provides a preliminary illustration of a training package for the Combat Battle Position Selection module of RPA (developed by McDonnell-Douglas Helicopter Systems Division, under contract to AATD). The training package is by no means either complete or final. For example, it lacks detailed scenarios that would be required to provide practice and feedback in the relevant concepts and strategies, and as a result focuses on explicit instruction more than the completed package would. Its primary purpose was to elicit feedback and comments that might lead to an improved package, and to establish the feasibility of training based on the principles laid out above.

We will describe the training requirements addressed by this package, the data gathering that guided the development of the training package, and the training package itself. Finally, we will describe the feedback that we have received from experienced pilots regarding the usefulness of this training.

Training Content

As noted in the previous chapter, five general training requirements or content areas have been identified for training users of decision aids. These methods are systematically related to elements of the trust model, as previously discussed. Our goal in this phase of the research was to seamlessly integrate three of these five methods into a single package:

1. The first training requirement conveys an accurate mental model of the aid, both through instruction regarding aid design and through direct experience with the aid. Both instruction and practice are designed to convey knowledge of the features that affect aid performance.
2. The second training requirement involves improving the user's skill in producing trust estimates and sensitizing users to the importance of estimating trust in the aid (for example, not expecting the aid to be always perfect).
3. The third training method teaches strategies for interacting with the aid, and improves user skill in determining which strategy is most appropriate on a given occasion.

The combination of these three methods should shape user performance in accordance with the event trees and benchmark models elaborated in the previous chapters.

Initial Data Collection and Design

In developing this training, we adopted a variety of methods for acquiring the information necessary to define the training content. Although time and availability of both pilots and aid designers was quite limited, the training development utilized many of the steps outlined earlier (in Chapter 0) for eliciting the content of mental models:

Interviews and observations with unaided, experienced decision makers. We observed pilots performing map exercises in which they identified combat battle positions manually, and conducted informal interviews with them afterwards.

Interviews and observations with decision makers using the aid. We observed pilots operating simulated versions of the Combat Battle Position Recommendation aid, and listened to discussions between the pilots and system engineers regarding their experiences with the aid. An important element of the observations was the occasional difficulty pilots had with some aspects of the system.

Discussions with decision aid designers, and examination of relevant documentation. We attended briefings at McDonnell-Douglas Helicopter Systems Division, Mesa, AZ, on the RPA Combat Battle Positions Recommendation (RPA/CBPR) aid, read relevant design documents, and had discussions with some of the aid designers and engineers. Our focus was on the version of the aid that existed at the time (April, 1997), despite our knowledge that modifications would continue to be made in its design. Prior to submitting the training for feedback, we made modifications to accommodate some of the changes in the RPA/CBPR system since our data collection visit.

Generating predictions regarding strengths and weaknesses of the decision aid. A key ingredient of the planned training was an attempt to convey a clear mental model of the aid, including an improved understanding of its strengths and weaknesses. A goal of the data collection, therefore, was to identify strengths and weaknesses of the RPA/CBPR aid. Potential problems with the aid were classified as either *gaps* or *unclear assumptions*. Gaps include evaluative criteria that the aid does not incorporate into its battle position selection process, but which are used by pilots in the manual performance of the task. In the version of the aid that we saw, there were a number of such omissions: e.g., lateral masking, cultural features, vegetation, and ingress/egress possibility, among others. We asked one of the pilots employed in testing RPA to assess the importance of some of the omitted factors on a scale of 0 to 100. The result is shown in Table 12.

It seems clear that some of these features, at least, have the potential to influence battle position selection significantly. Training might help pilots benefit from the aid while still incorporating these additional features into their thinking. Awareness of these omissions, and knowledge of strategies for compensating for them, might keep pilots from rejecting the aid altogether when it appears to produce inadequate recommendations.

Unclear assumptions include, for example, the way thresholds and weights are used by the aid, and the adaptive process by which thresholds are reduced when no successful option is found.[16]

---

[16] We incidentally gave some thought to the aid's current design. In some cases, design improvements would be quite beneficial, in addition to training. For example, with respect to adjusting thresholds and weights, at present these are accomplished on separate screens, with no feedback to the pilots regarding the impact of any changes that they make. As a result, this appeared to be highly non-intuitive, abstract task for the pilots, one they are not likely to engage in.

A substantial improvement would be to integrate the adjustment of thresholds and weights on the same screen, and to provide an immediate, dynamic display of the impact of any adjustment on the ranking of battle positions as shown on a map display. The current ranking of potential battle positions might be represented by numbers on a map (e.g., a "1" placed on the first ranking position, a "2" on the second place position, etc.). These numbers could change dynamically as thresholds and weights are changed by pilots. The pilots could thus observe concretely the impact of changing various factors in the current battlefield environment. They would be more likely to quickly find a set of parameters that agrees with their intuitive assessment of the battle positions.

There are many ways to adjust weights and thresholds. For example, weights on the different factors are naturally represented by the relative heights of bars, which users could drag up and down. The width of each bar would start out the same, but would be scaled (along the horizontal axis) to represent the range of possible scores on the factor in question. For example, the factor of proximity could start out with a range from 0 to 50 miles. Thresholds would be set by dragging one or both of the sides of the bar, reducing its width. So, to set a threshold of 10 for proximity, for example, the user would simply drag the right side of the bar from the point representing 50 to the point representing 10 (leaving a bar 1/5 the width of the original bar). This would mean that no battle positions farther than 10 miles away are to be accepted.

Such a display can neatly and intuitively represent the impact of a given factor on the evaluation of a particular battle position. It is the *area* of the bar whose height is the weight of the factor and whose width is determined by the distance on the horizontal axis from the threshold to the position's actual score. The *overall* evaluative score of any battle position will then be the *total* area of all the bars representing its scores on the various factors.

A second key ingredient of the training was strategies that users might employ to interact effectively with the aid. In this regard, we were guided by the theoretical framework described earlier, in conjunction with observations of pilots operating the system and performing the task manually.

In observing manual performance by two pilots, it seemed clear, at least from a behavioral standpoint, that the pilots used different strategies to identify suitable battle positions. One pilot seemed to examine promising positions serially, taking a fairly long time on each one, accepting or rejecting it, and then moving on to look for another promising candidate. The other pilot seemed to adopt a more breadth-first approach. He quickly drew a circle around the engagement area at the appropriate range, identified a large number of possible battle positions very quickly, and then picked the best from that fairly large number. It would be interesting to explore such potential individual differences further in the development of either training or the decision aid itself.

Table 12. Features omitted from an earlier version of the Combat Battle Position Selection aid, and their importance on a scale of 0 to 100 as assessed by an experienced Army pilot.

Feature	Importance
Lateral masking	65
Cultural features	60
Vegetation	95
Ingress/egress possibility	95
Vulnerability to unknown threats	80
Likely enemy avenue of approach	93
Multiple battle positions for team	90
Range and proximity for wingman	70
Terrain factors constraining threat and own firing	78

Illustrative Training Intervention for the Combat Position Selection Aid

Based on the data collection described above, and on the framework described in previous chapters, we developed a sample training intervention for user of the RPA/CBPR aid.

The training integrates the following features:

1. *Mental model of aid performance.* The training teaches the elements of a mental model of the RPA/CBPR aid. This includes both strengths and weaknesses of the aid: i.e., the features that the aid utilizes to evaluate battle positions, and the features that it currently omits (such as rotorwash, lateral masking, multiple battle positions for team members, and terrain and vegetation constraints on enemy location and own firing).

2. *Probabilistic assessment of trust.* The training sensitizes users to be explicitly aware of their degree of trust in the decision aid and the aid's fallibility. It introduces a concept of the aid as a team member, and sensitizes officers to dangers of over- and under-reliance on aid. A more complete version of this package would train users to estimate their degree of trust more accurately, by providing practice and feedback with respect to examples of aid recommendations under a variety of conditions in realistic scenarios.

3. *Strategies for user-decision aid interaction.* Finally, the CBPR training package exposes users to five strategies for interacting with the aid. These correspond to different degrees of reliance on automation (i.e., different automation modes). The five strategies are:

- Complete automation: Letting RPA do it.
- Letting RPA generate recommendations and then reviewing and critiquing them.
- Constraining RPA solutions , for example, by laying out exclusion zones within which battle positions will be unacceptable to the user.
- Changing the weights and thresholds that RPA uses to evaluate battle positions.
- Selecting battle positions manually, and using RPA to critique them. Two strategies for doing this were identified: by calling up the BRASSCRAF window, which evaluates any designated battle position in terms of the criteria implemented in the aid; and/or by allowing the aid to select its own battle positions and comparing its recommendations to one's own.

An important part of this training is helping users decide which strategy is appropriate under what conditions. As shown in the benchmark model in **Error! Reference source not found.**, this involves balancing time pressure,

degree of trust in the aid, and the cost of errors. In a more complete version of the package, the training will provide practice and feedback on strategy selection in the context of detailed scenarios.
A complete copy of the sample training package is included as Appendix B.

Evaluation of the Training Concepts

An informal test of these training concepts was conducted with the assistance of four highly experienced pilots or former pilots. We submitted the training package along with a questionnaire to the two pilots who served as subject matter experts at the previous Mesa exercise, to a consultant in the RPA program who is an experienced former pilot, and to the pilot who served as a subject matter expert at a second Mesa exercise. Thus, each participant in the evaluation had extensive RPA experience, and had served or is serving as a test subject for the RPA simulator studies or as a consultant to the project.
Each participant was given a copy of the training package and two questionnaires, one concerning their trust in the aid, and the other concerning their attitude toward the training.[17] In the first questionnaire, participants were asked for the following information:

- How much do you trust this aid (0-100 scale)?
- Why? What influences your degree of trust?
- Will you use this aid? Under what conditions will you use it? Under what conditions will you not use it?

In the second questionnaire, participants were asked to evaluate the training:

Q1. Did training increase your understanding of the aid?
Q.2 Did training change your confidence or trust in the aid?
Q3. Is the training likely to influence how much you use the aid?
Q4. Is this training likely to influence how you use this aid?

In addition, participants were invited to make other general or specific comments on the training. The following is the complete text of the answers to the four evaluative questions regarding the training and additional comments by the participants. We refer to the four participants as P-1, P-2, P-3, and P-4, respectively. Note that not all participants answered all questions.

General Comments

P-1: Overall good idea. I agree that we shouldn't try to sell what isn't there [in the aid]. This is a systematic approach to ways of using it – and it's good.
The training presupposes some familiarity with the system. Would be good for pilots who know nothing about the aid, but would need more introduction.
P-2: Really liked it. A super hand out. Something like this is the way I would like to start out learning RPA.
P-3: This appears to be a very understandable and linear format for training an individual on the implementation and tactical usage of RPA. The overall approach is excellent.

Q1. Did training increase your understanding of the aid?

P-1: A lot. Story boards and exercises would be very beneficial.
P-2: Yes. Step by step process — allows you to look at all the pieces. Points out the good and the weaknesses. Gives you a better overall concept.
P-3: Moderately. Effects / impacts of weighting / thresholding are not covered completely.
P-4: No. This approach would be valuable if it spoke the same language as current training. I like the idea of RPA as a crew member. But need to tie in with current Crew Coordination training.

Q2. Did training change your confidence or trust in the aid?

P-1: No. It gave me more ideas on how to use it.
P-2: Yes. A more global view of RPA. Previous training was only on the system. Didn't highlight strengths and weaknesses.

Q3. Is the training likely to influence how much you use the aid?

P-1: Yes. See above. If it provides new or different ways to use it, it will increase how much.
P-2: Absolutely. With better understanding, gives me flexibility to utilize strengths and minimize its weaknesses.

Q4. Is this training likely to influence how you use this aid?

P-1: Yes! As a guide or a starting point.
P-2: Yes. Use strengths, be aware of weaknesses. Better able to tweak the system to my desired outcome or results.

[17] Our original plan was to have the participants complete the questionnaire regarding trust in the aid both before and after the we conducted training. Due to limitations of time at the Mesa exercises, however, the participants completed the questionnaires only after examining the training materials. Two participants gave their answers to the questionnaires orally.

P-3: Delineation of RPA and pilot strengths / weaknesses develops the proper user expectation of system performance and required interface.

The comments were generally quite positive, at least for three of the four pilots. Most of the criticisms (not enough story boards and exercises; incomplete explanation of thresholds and weighting; incomplete integration with crew coordination training; presupposing some familiarity with the aid) can be attributed to the brevity of the material and the constraints on the development process at this phase of the research. A fuller version of this training will cover more aspects of the aid, will include detailed scenarios in which pilots can practice and receive feedback regarding user-aid interaction skills, and will be better integrated with other training that pilots receive.

To what degree did the training succeed in accomplishing its primary goals? These goals were: to convey an improved mental model of the aid, to sharpen assessments of trust in the aid, and to train strategies for interacting with the aid? Table 13 tabulates the responses that were received to the four questions about the training. Some very preliminary conclusions, based on this very small sample, are the following:

- The training appears to have been somewhat successful in conveying improved mental models, based on three out four positive responses to question 1.
- The training may have been less successful in sharpening trust in the aid, based on mixed responses to question 2. This may be due in part at least to the very limited experience these pilots had with the aid prior to training, and to the lack of actual practice provided by the training.
- The training was probably most successful in teaching strategies for interacting with the aid, based on three positive responses to question 4.

Table 13. Responses to evaluative questions about the training.

Question	**Yes**	**No**
Q1. Increase understanding of aid?	3	1
Q2. Change trust in aid?	1	1
Q3. Change how much use aid?	2	0
Q.4 Change how use aid?	3	0

Another way of looking at the responses is to ask how individual pilots reacted to the different goals. For example, some participants may have responded more to the goal of increasing understanding in the aid, while others responded more to the goal of learning new strategies for interacting with the aid. Some pilots, on the other hand, may have responded to both goals. Table 14 lists the answers for each participant that referred to either one of these two goals: increased understanding of the aid and learning better ways of using the aid.

All three of the pilots who responded favorably to the training made reference to both of the goals. Moreover, two of these pilots explicitly linked the two goals in their answers to the same question: Participant 2 in response to question 3, and participant 3 in response to question 4. These pilots appear to agree that improved mental models of the aids can provide a foundation for better user-decision aid interactions strategies.

Table 14. Tabulation of questions in response to which participants referred to different goals of the training.

	Goal Referred to by Participant	
Participant	**Understanding strengths and weaknesses**	**Learning ways of using the aid**
P-1	Q1	Q2, Q3
P-2	Q1, Q2, Q3	Q3
P-3	Q4	Q1, Q4
P-4		

SITUATION AWARENESS MEASURES

This section presents a model of situation awareness (SA) in the context of the Rotorcraft Pilot Associate (RPA) for attack helicopter missions. We present an operational model of SA that is comprised of knowledge of the critical elements of the situation and the values of these elements as a function of time. Based on this model we discuss the measurement framework we are using to identify the critical elements of SA for this domain and to investigate the development of the pilot's SA over time and its effect on RPA-aided mission performance.

Introduction

Performance in complex military systems is often based on the integration of human observations and judgments with automated information. For example, understanding and using the RPA to locate combat battle position can be

a cognitively complex task. In this context, the pilot's judgment is based on a combination of visual, contextual, and automated information. The level of RPA-aided pilot performance that can be achieved is a function of the quality of the various sources of information, the trust that the pilot has in the RPA, and the pilot's ability to merge together effectively information and/or decisions from those sources.

A critical advantage of human operators over automated systems is their ability to make use of *situational information* in performing their tasks. Humans are efficient and robust information processors in that they are able to use what they *expect* to see or hear to interpret incomplete and uncertain incoming information. On the other hand, humans can commit serious errors when they make decisions based on their expectations, as has been tragically illustrated by incidents in which soldiers fired at friendly units in locations where they had expected the enemy to be present.

Aviation records suggest that many pilot-caused errors result from a lack of situation awareness (SA). Nagel (1988) notes that breakdowns in SA are one of the most serious problems in aviation operations. Hartel, Smith, and Prince (1991) report that in a Navy and Marine analysis of mishaps, lack of SA was the most frequently cited causal factor. Thornton, Kaempf, Zeller, and McAnulty (1991) found that lack of relevant and timely information was related to Army tactical helicopter crews' poor performance in navigation and threat evasion. Both observational and experimental data support the intuitive assumption that higher levels of SA lead to better task performance, but the quantitative nature of this relationship has not been established.

This report addresses SA in the context of RPA-aided pilot performance. A key assumption is that trust in the RPA aid is a function of the situation features. Therefore, a central premise to this approach is that better SA will increase a pilot's ability to apply his/her situational knowledge to integrate better the planning and decision-making recommendations of an RPA module with his/her own decisions, thereby leading to improved pilot-system performance. The key objectives of this work are to demonstrate empirically the relationship between level of SA and trust in RPA, and to recommend principles for the display of SA information that optimize overall performance. To achieve these objectives, we formulate an operational definition of SA in the RPA domain and devise a method to measure quantitatively an operator's level of SA.

What is Situation Awareness?

Although SA is an appealing and widely-used term, there is no universally accepted definition of the term, and no standard methodology for defining the elements of SA or assessing an individual's level of SA in the context of a particular task domain. Wellens (1993) suggests that in a military context, SA can be "roughly conceived of as an individual's internal model of the world at any point in time." The most widely referenced definition is one proposed by Endsley (1988) who defines SA as "the perception of the elements in the environment within a volume of time and space, the comprehension of their meaning, and the projection of their status in the near future."

As a psychological construct SA has been discussed as both process and product. Like Endsley (1995a), we view SA as a state of knowledge captured at a particular moment in time, and situation assessment as the process of acquiring or maintaining SA. Clearly an individual's level of SA can change over time, and is dependent upon the amount and quality of information that is available–which is independent of the particular individual–and the individual's ability to perceive and comprehend that information in a timely fashion–which is a function of the individual's prior knowledge about what elements are important, how they are relevant, and how they change over time (Adams, Tenney, and Pew, 1995). Endsley's definition embraces both of these aspects of SA, with "perception" implying information availability and acquisition, and "comprehension and projection" suggesting the individual's ability to use the information.

The assessment of an individual's level of SA has been an equally elusive problem. (See Endsley, 1995b, and Adams et al., 1995 for a general discussion of the measurement of SA.) Because SA is a vaguely-defined concept and the elements of SA are not easy to identify or to quantify, it has sometimes been assessed in terms of task performance. However, as Endsley (1995a) points out, superior performance can result in spite of poor SA, and likewise high levels of SA do not always result in superior performance. In order for the concept to have meaning, SA must be defined and measured *independently* of performance. For example, we cannot directly infer a pilot's level of SA by measuring aspects of his or her flying performance. Rather, we want to identify and measure those elements in the situation which are *predictive* of performance.

Although there is no agreed-upon definition for SA, researchers do agree that it must be defined in the context of a particular task. Moreover, for the concept to have meaning, one must be able to specify the elements that comprise SA, with particular sets of elements being relevant for particular system states. Furthermore, although the elements of the situation may change dynamically over time, only some of the changes will be large or severe enough to cause a change in the situation from the point of view of the system operator (Pew, 1994). For some elements, there may be certain ranges in which knowledge of the precise value of the element is not critical for SA, and other ranges in which the precise value is critical.

In order to investigate the relationship between SA and aided decision-making performance, we need an operational definition of SA in the helicopter domain and a way to measure an individual's level of SA.

Modeling Framework

We propose a cognition-based, process-oriented research framework for establishing the elements of SA in the context of RPA-aided pilot performance, assessing an individual's level of SA, and investigating quantitatively the relationship between SA and task performance.

Figure 29 depicts a two-stage process by which battle position decision-making performance is carried out. The first process embodies the evolution of SA. Situation assessment is a dynamic process, with input to the process coming from the individual decision maker's background knowledge and experience, and from global and local elements in the situation. The global elements, encompassing such factors as the geopolitical situation, establish the general situation and determine the local factors that will comprise the critical elements of SA. The global factors are the 'givens' that interact with prior knowledge and experience to provide an individual with an initial mental model of the situation which, in turn, helps determine the information that is critical for SA in this particular situation. The output of situation assessment is a level of SA, which can be measured at any point in time.

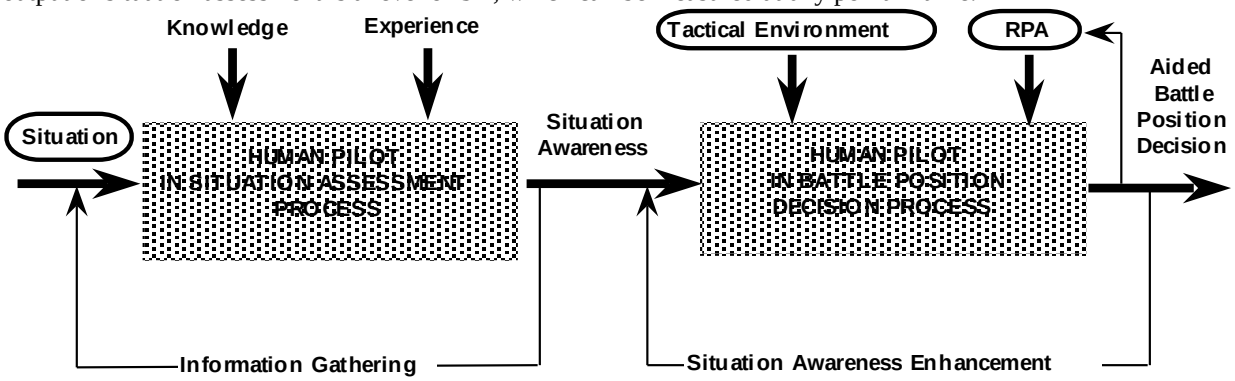

Figure 29. Process Model of Dynamic Situation Assessment and RPA-Aided Decision-making

The second process in Figure 29 concerns the decision-making process of interest, i.e., in this case, the RPA-supported battle position decision process. However, this second process could be any of several decision processes occurring—sometimes simultaneously—in the cockpit. However, note that the decision objectives may drive, to a great extent, the nature of the elements of the situation the pilot must be aware of. The decision making process takes as its input the individual's level of SA (measured as the output of the first process), information about the tactical environment, and information provided by an ATR. The output is a final decision on how to locate best combat battle positions. Note that the output of this process becomes an input to the situation assessment process, and can affect the individual's subsequent level of SA.

Modeling Situation Awareness

An individual's level of SA is a dynamic variable, evolving over time as the situation changes. Since SA is domain-dependent, what information is perceived and how it is interpreted is a function of the individual's task objectives. We propose that from an operational point of view SA is comprised of knowledge of four aspects of the situation. These four dimensions incorporate the integrated processes of perception, comprehension, and projection over space and time proposed by Endsley (1988), albeit organized in a slightly different manner:

- What are the critical variables in the situation now? (comprehension)
- What are the current values (or states) of these variables? (perception)
- What will be the critical variables in the future? (comprehension/projection)
- What will their values be? (projection)

The complexity inherent in the concept of SA stems in part from the fact that these four aspects are interdependent, and not necessarily sequential. In fact one might argue that one would seek information ("perception") that one needs for the situation/decision at hand ("comprehension"). Moreover, knowledge of the critical variables and of the interrelationships among variables directs information gathering ("perception"), which, in turn may influence both comprehension and projection. Projection may direct perception and comprehension in the future.

The attention to the critical elements of the situation (the first and third aspects) suggests that not all the situational elements are of equal importance. For example, air-to-air combat fighters view information about enemy aircraft as more important than information about friendly aircraft (Endsley, 1993). Furthermore, the same informational

element may not be equally important over time. For locating an ideal combat battle position, knowledge of the location of an enemy aircraft may become more important over time as the pilot's own aircraft approaches the enemy aircraft. The task of an individual who is trying to maintain a high level of SA is not to represent the whole state at any time, but only the critical elements, the ones that will affect mission performance.

We represent the relationship between SA and performance (P) by a *performance sensitivity model*:

[1] $P(t) = P(S_i, \text{other non-SA factors})$

thus,

[2] $\Delta P(t) = e_1\Delta S_1 + e_2\Delta S_2 \ldots + e_n\Delta S_n$

where:

P = mission performance

ΔP = decrement in mission performance due to "less-than-perfect" SA

f = functional relationship

S_i = element in the situation vector

ΔS_i= decrement in accuracy of estimate of value of s_i

e_i = sensitivity coefficient or criticality factor for s_i

t = time

This model represents the relationship between elements of the situation and mission performance in terms of the extent to which a decrement in the accuracy of an estimate of particular elements of the situation is related to a decrement in mission performance. The e_i values reflect the sensitivity (or criticality) of each element for performance. The magnitudes of these sensitivity coefficients are determined by the mission itself, not by the particular individual who is acting in the situation. They are dynamic in that their level of criticality may change over the course of the mission (for example, the weather may be especially critical during a particular phase of a mission). Knowledge of the critical elements is derived from an individual's past experience and his or her mental model of the situation. Studies of decision making expertise in complex task domains (MacMillan, Entin, and Serfaty, 1993) indicate that experts tend to agree on which elements are critical and which ones are less so.

The second and fourth aspects of SA concern the perception of the current values of situational elements and projection of their future values. Accuracy in estimating the values of the situational elements (ΔS_i) is dependent upon the operator's ability to perceive or infer the current values of the elements. Clearly the rate of change of the values of the situational elements is dependent upon the element (e.g., the position of an aircraft will change more rapidly than that of a tank) and the particular scenario as it unfolds (e.g., on some days the weather conditions will change more rapidly than on other days). The rate and extent of change in the values of the situational elements is also determined by actions taken by the individuals involved in that mission.

We represent the relationship between previous and current values of the elements by a *dynamic situation model*:

[3] $Si(t) = g\,[Si(t-1), d(t-1), e(t)]$

where:

Si(t) represents the values of the situational elements at the current time

g [ ] represents the structure of the pilot's dynamic mental model of the situation

Si(t-1) represents the values of the situational elements at the previous point in time

d(t-1) represents actions taken at the previous time (e.g., a change in heading)

e(t) reflects the inherent process uncertainties (e.g., random flux in wind speed)

An individual can obtain the values for the elements of the situation through two mechanisms: direct observation and estimation. For those variables that are observable (for example, airspeed or cloud cover), the individual can perceive the information directly. But the values of some elements may not be directly available. For those elements that cannot be observed, the individual must reconstruct or estimate their value based on a mental model of the situation, current observations of other, related elements, and (if they are known) previous values of those elements. We can think of the function g as in part embodying an individual's dynamic mental model of how, in the absence of external forces, the elements are related to one another and how they change over time. For example, if the location of an enemy aircraft cannot be observed at a particular time, an individual can infer its location based on his knowledge of how fast that type of aircraft travels, atmospheric conditions such as wind speed, and the location of that aircraft at time t-1. This knowledge is embodied in the individual's dynamic mental model of the situation.

In reality it is not possible for an individual to have perfect SA at any time. In part, for information that is directly observable, the shortfall may be attributable to observation error. The individual may, for example, misread the airspeed or misunderstand the location of an enemy unit. The magnitude of the observation error will be related to the quality of the information that is available and the individual's observation skill. In part the shortfall may be due

to faults in the individual's mental model of the situation. For example, in estimating the location of an enemy unit that is not directly observable, an individual may use a faulty mental model of the enemy's scheme of maneuvers. For situational elements whose values must be inferred from previous values, Equation 3 indicates that accuracy in estimating the current values will depend on the accuracy of the previous estimates of the values of those elements. In part the shortfall in SA may be due to information that is not available and cannot be reliably inferred from other information.

The third and fourth aspects of SA involve projection: what will be the critical variables in the future and what will their values be? Knowledge of these aspects must be based on the individual's dynamic model of how the situation will evolve in the future. Just as the individual's comprehension of the current situation directs his search for current situational information, his mental model of the current situation will direct his projection of the future situation, both in terms of what the critical elements will be and what their values will be.

Thus, an individual's level of SA depends upon his or her knowledge of the critical elements and the degree to which he or she is able to correctly perceive or infer the values of the critical elements of the situation over time. An individual who has accurate estimates of the sensitivity coefficients, who perceives the available information accurately, and who has an accurate dynamic model of the situation (that is, an individual who accurately reconstructs or infers unobservable information at the current time and uses that model to predict future values of critical elements) will have a high level of SA. Equation 2 indicates that an individual whose estimates of the values of critical elements are perfectly accurate, but whose estimates of the values of non-critical elements are highly inaccurate will suffer little decrement in performance, whereas an individual whose estimates of all elements are only slightly inaccurate may actually suffer a greater decrement in performance.

To test the relationships described in Equations 1, 2, and 3 we must measure SA as it develops over time. A typical attack helicopter mission is organized into five phases: (1) planning, (2) preparation, (3) ingress, (4) battle position, and (5) egress. We can use these five phases as a way of operationalizing time. As shown in Fig. 3, to see how SA develops over time, we can measure SA at each phase of the mission. To gain additional information about how projection contributes to both the formation of SA and to performance, we can measure an individual's *estimated* level of SA for Phase 4, the phase in which the performance task occurs, at each of the three preceding phases. In Figure 30 we represent SA at each phase by $SA_{(i)}$, the estimates of SA for Phase 4 at each of the preceding phases by $SA_{e4(i)}$ (i.e., "you are now in phase 1; what is your assessment of what the situation will be in phase 4"), and the performance at Phase 4 by P_4. If projection is indeed an important aspect of SA, then we would expect a positive relationship between the estimates of SA for Phase 4 at the previous phases of the mission and the actual level of SA at Phase 4.

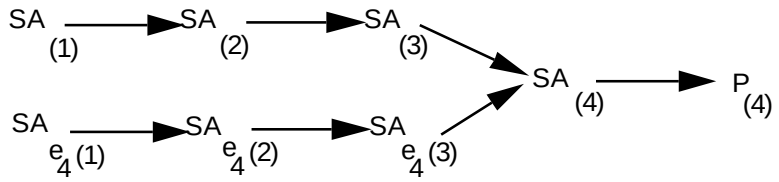

Figure 30. Measurement Plan for Assessing Current and Projected SA.

Details on the operational implementation and testing of this measurement approach will be described in future research. In the next section we suggest a method for identifying the elements of the situation and the values of the sensitivity coefficients.

Identifying and Measuring the Critical Elements of the Situation

Using the conceptual approach described above, the first goal is to identify the critical elements of SA for attack helicopters missions. A three-step process is used to identify the critical elements of SA for mission performance, or, in a more restrictive case, for performance related to location of combat battle positions. The could apply to parts of the mission, or the mission as a whole.

The first step is the identification of a candidate set of situational elements (S_i's) that may affect performance at various times (or stages) of the mission. Based of reviews of documents on tactics, techniques and procedures for attack helicopters (FM1-112), we have identified an initial candidate set of local elements of SA for this type of attack scenario. The candidate set of elements is presented in Table 15.

Table 15. Elements of the Situation for Attack Helicopter Pilots

Task Organization
Current Situation:
  Friendly: number, type location of units
  Enemy: number, type location of units
  Weather: temperature, visibility, cloud types
Mission
Execution:
  Concept of Operations:
  Instructions/changes
  Scheme of Maneuver
  Friendly Art/CAS
  Enemy ADA Support
  Coordinating Instructions:
  Action on Contact
  Critical Times
  Report Requirements
  Flight Coordination
  Special Mission Equipment
	Service Support
	Commands/Signals
	Aircraft Status
	Crew Status
	Armament System: Status/Operations
	Sighting Subsystems: Status/Operations
	Aircraft Survivability Equipment: Status/Warnings
	Communications System: Status/Operations
	Fuel System: Status/Operations
	Navigation System: Status/Operations
	Crew Briefing
	Changes to Planning Information

The second stage in this procedure is to obtain estimates of the criticality of these elements at the various stages the mission–in other words, estimates of the e_i values as a function of time. To accomplish this we should conduct structured, scenario-based interviews with experienced attack helicopter pilots. The pilots would be asked to rate the criticality of the elements enumerated in Table 15. They would also be asked to specify other elements that they believe are critical to determine optimal battle position performance. Assessments of the criticality of various elements will be made in each phase of the mission, so we can capture the criticality of the elements as a function of time. In addition to describing and evaluating the elements, the pilots would be asked to explain how that information is used, and how frequently it is updated. Their responses will provide descriptive information about how experts use projection as an aspect of SA.

In a third stage the criticality ratings obtained from the experienced pilots are integrated to derive a set of criticality weights (sensitivity coefficients) as a function of time. Based on interviews with expert pilots, we will also have information about how each element of information is used, how the elements are interrelated, and how accurate an estimate of the value of the element is needed, as a function of time. For example, in the ingress stage of a mission, it may be sufficient to know the approximate location of enemy units, whereas in the battle position stage, a more precise determination may be needed. This information is needed in Equation 2, in order to prescribe meaningful units of change.

Once the critical elements of SA have been identified, we will use a scenario-based method for assessing an individual's knowledge of the criticality and value of each element at various stages in a mission and empirically relating those measures to aided performance. By comparing the individual's reports of the criticality of individual elements to the values obtained from experts and estimates of the values of the elements to their actual values, we can evaluate the extent to which knowing which elements are critical and knowing what the values of the elements are contribute to high levels of SA. Comparison of an individual's projections of the importance and value of elements in the future to the weighting and value he or she actually gives them at that future time will contribute to our understanding of how projection contributes to high levels of SA and to what extent the comprehension of the current situation and projection of the future SA contribute to performance.

In summary, in order to assess how SA affects trust in automated aids such as the RPA, we must first identify the key situational variables that affect performance in this domain. To demonstrate real design value, one must operationalize SA in a quantifiable manner. Therefore, it becomes important to develop theory-based, unambiguous, objective, and quantifiable metrics that can predict trust and use in RPA as well as subsequent human-system performance improvement.

Workload Measures

Mental workload is an intervening variable between inputs to the human-machine system and performance outcomes and errors. Unfortunately workload is not directly measurable from the system observable variables. Moreover the relationship between workload and performance is generally not a simple (e.g., linear) one.

There are three main ways to infer or assess workload in cognitively complex tasks:

- Subjective measures: Direct subjective report of mental workload using rating scales (see below)

- Performance-based measures: Secondary task performance measures, in which a additional task (e.g., tracking) is superimposed to the main task. Performance decrement on the secondary task is an indicator of the workload generated by the primary task.
- Physiological measures: Several indices taken in combination, such as heart rate variability, pupillary diameter, galvanic skin response, evoked potentials, etc...

Despite a wealth of research there is no consensus on either definitions or agreement on the best measurement methods of mental workload. In research efforts in which it is important to minimize intrusion into the main task—this project falls in this category—, we have found that subjective measurement methods work best while providing both ease-of-use and reliability.

Two measures of mental workload have been extensively used in cognitively-demanding task contexts: the Subjective Workload Assessment Technique (SWAT) and the NASA Task Load Index (TLX). The SWAT (Reid, Singledecker, Nygren, & Eggemeir, 1981) uses three dimensions of workload: mental effort, time demand, and stress. The TLX (Hart & Staveland, 1988) has six dimensions: the first three (mental demand, physical demand, and temporal demand) are viewed as relating to the demands imposed on the subject and the other three (performance, effort, and frustration level) to the interactions of a subject with the task.

Both measures involve a procedure by which the workload dimensions are calibrated to an individual's perception of the most relevant dimensions for a particular type of task. We recommend to use the TLX for two reasons. First, it requires less time from the subject than the SWAT to administer the calibration ratings, and it involves very little post-processing. In addition, the six TLX subscales provide more specific diagnostic information about the sources of workload than does the SWAT. The TLX can be administered at the end of a scenario, or in a retrospective but less reliable manner, by the subject providing a posteriori evaluation of the mental workload based on a recollection of events or on an evaluation of a mission audio- or video-recording.

Secondary task measures of workload involve adding a concurrent task (such as monitoring for a randomly presented signal) to the task of interest, and instructing the subject to allocate as much capacity as required to the primary task and only what is "left over" to the concurrent task. Performance in the secondary task is then measured and reflects the time-varying capacity demands of the primary task.

Problems with the secondary task method are two-fold: First, despite the instructions to allocate it only left over capacity, it can be intrusive with respect to the way the primary task is performed. The second problem is that concurrent task performance sometimes appears insensitive to changes in primary task demands. The latter might be explained in two ways: First, lack of effect on the secondary task would be expected if the two tasks do not overlap in their demands for resources. Wickens (1984) and others (e.g., Cohen, 1979) have proposed multiple-resource models, of capacity in which, in essence, all resources are specialized. A second possible cause of the lack of sensitivity, however, is more troublesome. Subjects might be shifting resources to the secondary task at the expense of the primary task, contrary to the instructions. That this appears to be the case is suggested by the evidence of intrusion on primary task performance referred to earlier. Such uncontrolled shifting of resources would invalidate the technique.

One solution to this problem was offered by Sperling (1978). He advocates varying the relative rewards for the two concurrent tasks. A plot of the performance of one task against performance on the other is called an Attention Operating Characteristic or, alternatively, a Performance Operating Characteristic. The AOC is analogous to the Receiver Operating Characteristic (ROC) in signal detection theory, and serves an analogous purpose, of unconfounding decision criteria from underlying capabilities. The curve traced out by the two tasks represents their shared capacity. The more area under the curve, the more independent the two tasks are, i.e., the less capacity they compete for. The closer the slope comes to 1, the more intensely they compete, i.e., any gain in performance by one task is paid for by a commensurate loss in performance by the other task. The particular point on this curve that the subject adopts in any particular condition is less important; it reflects his perception of the relative costs and rewards of the two tasks.

Unfortunately, this solution has its own problems: It multiplies the number of conditions that must be tested; moreover, the ability to diagnose time-varying changes in capacity demands is sacrificed. A more pragmatic solution is to take the traditional secondary task technique and mitigate its potential biases. We propose to do this by (1) ensuring that the secondary task is a realistic part of the real-world scenario, and that its secondary character is reinforced by the context of the scenario, and (2) by carefully measuring performance in the primary task to ensure that it in fact is not affected by the secondary task.

Our plan is to apply a measurement battery with two tiers: A subjective multidimensional questionnaire administered after the task, to capture the subject's perceptions of the global workload in the task; and a secondary task to be applied throughout the primary task, to assess moment-by-moment fluctuations in workload as well as

potential resource-specific effects. The combination of these two methods should help us trace the decision making processes of users of decision aids as they respond to dynamic task demands.

REFERENCES

Adams, M., Tenney, E., and Pew, R. 1995. Situation awareness and the cognitive management of complex systems. *Journal of the Human Factors and Ergonomics Society, 37(*1):85-104.

Andes, R.C., and Rouse, W.B. 1992. Specification of adaptive aiding systems. *Information and Decision Technologies, 18*:195-207.

Barber, B. 1983. *The logic and limits of trust*. New Brunswick, NJ: Rutgers University Press.

Billings, C., and Woods, D. 1994. Concerns about adaptive automation in aviation systems. In *Human performance in automated systems: Current research and trends*, ed. M. Mouloua and R. Parasuraman, 264-269. Hillsdale, NJ: Erlbaum.

Brown, R.V. 1971. *Research and the credibility of estimates.* Homewood, IL: Richard D. Irwin, Inc.

Brown, R.V., Kahr, A.S., and Peterson, C.R. 1974. *Decision analysis for the manager*. NY: Holt, Rinehart, and Winston.

Cohen, M.S. 1979. *Voluntary control over specialized resources*. Doctoral dissertation. Harvard University.

Cohen, M.S. 1986. An expert system framework for non-monotonic reasoning about probabilistic assumptions. In *Uncertainty in artificial intelligence,* ed. J.F. Lemmer and L.N. Kanal. Amsterdam: North Holland Publishing Co.

Cohen, M.S. 1993. The naturalistic basis of decision biases. In *Decision making in action,* ed. G.A. Klein. Norwood, NJ: Ablex.

Cohen, M.S., and Brown, R.V. 1980. *Decision support for attack submarine commanders* (Technical Report 80-11). Falls Church, VA: Decision Science Consortium, Inc.

Cohen, M.S., and Freeling, A.N.S. 1981. *The impact of information on decisions: Command and control system evaluation* (Technical Report 81-1). Falls Church, VA: Decision Science Consortium, Inc.

Cohen, M.S., and Hull, K. 1990. *Recognition and metacognition in sonar classification* (Technical Report). Reston, VA: Decision Science Consortium, Inc.

Cohen, M.S., Laskey, K.B., and Tolcott, M.A. 1986. *A personalized and prescriptive decision aid for choice from a database of options* (Technical Report 86-1). Falls Church, VA: Decision Science Consortium, Inc.

Cohen, Marvin S. & Freeman, Jared T. Thinking naturally about uncertainty. In *Proceedings of the Human Factors & Ergonomics Society, 40 Annual Meeting*. Santa Monica, CA: Human Factors Society, 1996.

Cohen, M.S., Freeman, J.T., and Thompson, B.B. 1997.Critical thinking skills in tactical decision making: A model and a training strategy. In *Decision making under stress: Implications for training and simulation,* ed. A. Cannon-Bowers and E. Salas. Washington, DC: American Psychological Association.

Cohen, M.S., Thompson, B.B., & Freeman, J.T. 1997. *Cognitive aspects of automated target recognition interface design: An experimental analysis.* Arlington, VA: Cognitive Technologies, Inc.

Duncan, P. C., Rouse, W., Johnston, J. H., Cannon-Bowers, J. A., Salas, E., & Burns, J . 1996. Training teams working in complex systems: A mental model-based approach. *Human/Technology Interaction in Complex Systems, 8*: 173-231.

Einhorn, H.J., and Hogarth, R.M. 1984. Ambiguity and uncertainty in probabilistic inference. *Psychological Review*.

Endsley, M. 1988. Situation awareness global assessment technique SAGAT. In *Proceedings of the National Aerospace and Electronics Conference,* 789-795. New York: IEEE.

Endsley, M. 1993 A survey of situation awareness requirements in air-to-air combat fighters. *International Journal of Aviation Psychology*, *3:*157-168.

Endsley, M. 1995a. Toward a theory of situation awareness in dynamic systems. *Journal of the Human Factors and Ergonomics Society, 37*(1):32-64.

Endsley, M. 1995b. Measurement of situation awareness in dynamic systems. *Journal of the Human Factors and Ergonomics Society, 37*(1):65-84.

Endsley, M.R. 1996. Automation and situation awareness. In *Automation and human performance: Theory and applications,* ed. R. Parasuraman and M. Mouloua. Mahwah, NJ: Lawrence Erlbaum Associates.

Endsley, M., and Kiris, E. 1994. The out-of-loop performance problem: Impact of level of automation and situation awareness. In *Human performance in automated systems: Current research and trends*, ed. by M. Mouloua and R. Parasuraman, 50-56. Hillsdale, NJ: Erlbaum.

Goodman, B. 1972. Action selection and likelihood estimation by individuals and groups. *Organizational Behavior and Human Performance, 7:*121-141.

Hart, S.G. and Staveland, L.E. 1988 Development of the NASA-TLX Task Load Index: Results of empirical and theoretical research. In *Human mental workload,* ed. P.A. Hancock and N. Meshkati. Amsterdam: Elsevier.

Hartel, C., Smith, K., and Prince, C. 1991. *Defining aircrew coordination: searching mishaps for meaning*. Paper presented at the Sixth International Symposium on Aviation Psychology, Columbus: The Ohio State University.

Kadane, J.B., and Lichtenstein, S. 1982. *A subjectivist view of calibration* (Report 82-6). Eugene, OR: Decision Research.

Kahneman, D., Slovic, and Tversky, A. 1982. *Judgment under uncertainty: Heuristics and biases*. NY: Cambridge University Press.

LaValle, I.H. 1968. On cash equivalents and information evaluation in decisions under uncertainty. *American Statistical Association Journal, 63*:252-290.

Lee, J.D., and Moray, N. 1994. Trust, self-confidence, and operators' adaptation to automation. *International Journal of Human-Computer Studies, 40:*153-184.

Lee, J.D., and Moray, N. 1992. Trust, control strategies, and allocation of function in human-machine systems. *Ergonomics, 35*(10):1243-1270.

MacMillan, J., Entin, E.B., and Serfaty, D. 1993. Evaluating expertise in a complex domain–Measures based on theory. In *Proceedings of the Human Factors and Ergonomics Society 37th Annual Meeting,* 1152-1155. Santa Monica: Human Factors and Ergonomics Society.

Mosier, K.L., and Skitka, L.J. 1996. Human decision makers and automated decision aids: Made for each other. In *Automation and human performance: Theory and applications,* ed. R. Parasuraman and M. Mouloua, 201-220.Mahwah, NJ: Lawrence Erlbaum Associates.

Muir, B.M. 1987. Trust between humans and machines, and the design of decision aids. *International Journal of Man-Machine Studies, 27*:527-539.

Muir, B.M. 1988. Trust between humans and machines, and the design of decision aides. In *Cognitive engineering in complex dynamic worlds* , ed. E. Holnagel, G. Mancini, and D.D. Woods. London: Academic Press.

Muir, B.M. 1994. Trust in automation. Part I. Theoretical issues in the study of trust and human intervention in automated systems. *Ergonomics, 37*(11):1905-1922.

Muir, B., and Moray, N. 1987. Operator's trust in relation to system faults. *IEEE International Conference on Systems, Man, and Cybernetics*, 258-263. Alexandria, VA.

Muir, B., and Moray, N. 1996. Trust in automation. Part II: Experimental studies of trust and human intervention in a process control simulation. *Ergonomics, 39*(3):429-460.

Nagel, D. C. 1988. Human error in aviation operations. In *Human factors in aviation,* ed. E. L. Weiner and D. Nagel. San Diego, CA: Academic Press.

Nisbett, R., and Ross, L. 1980. *Human inference: Strategies and shortcomings of social judgment*. Englewood Cliffs, NJ: Prentice-Hall, Inc.

Parasuraman, R., and Mouloua, M. 1996. *Automation and human performance: Theory and applications*. Hillsdale, NJ: Erlbaum.

Parasuraman, R., and Riley, V. 1997. Humans and automation: Use, misuse, disuse, abuse. *Human Factors*.

Parasuraman, R., Molloy, R., and Singh, I.L. 1993. Performance consequences of automation-induced "complacency*." The International Journal of Aviation Psychology, 3*:1-23.

Parasuraman, R., Mouloua, M., and Molloy, R. 1996. Effects of adaptive task allocation on monitoring of automated systems. *Human Factors, 38*:665-679.

Perrow, P. 1984. *Normal accidents*. NY: Basic Books.

Pew, R. 1994. An introduction to the concept of situation awareness. In *Situational awareness in complex systems,* ed. R. D. Gilson, D. J. Garland, and J.M. Koonce. Daytona Beach, FL: Embry-Riddle Aeronautical University Press.

Raiffa, H. & Schlaifer, R. 1961. *Applied statistical decision theory.* Cambridge, MA: The M.I.T. Press.

Rasmussen, J. 1983. Skills, rules, and knowledge: Signals, signs, and symbols, and other distinctions in human performance models. *IEEE Transactions on Systems, Man, and Cybernetics, SMC-13(*3):257-266.

Reason, J. 1990. *Human error*. Cambridge, UK: Cambridge University Press.

Reid, G. B., Singledecker, C.A., Nygren, T.E. and Eggemeir, F.T. 1981. Development of Multidimensional subjective measures of workload. *Proceedings of the 1981 IEEE International Conference on Cybernetics and Society*, 85-104. Atlanta, GA.

Rempel, J. K., Holmes, J.G., and Zanna, M.P. 1985. Trust in close relationships. *Journal of Personality and Social Psychology, 49*:95-112.

Riley, V. 1989. A general model of mixed-initiative human-machine systems. *Proceedings of the Human Factors Society 33rd Annual Meeting—1989,* 124-128. Minneapolis, MN.
Riley, V. 1994. A theory of operator reliance on automation. In *Human performance in automated systems: Current research and trends,* ed. M. Mouloua and R. Parasuraman, 8-14. Hillsdale, NJ: Erlbaum.
Roth, E.M., Bennett, K.B., and Woods, D.D. 1987. Human interaction with an "intelligent" machine. *International Journal of Man-Machine Studies, 31*:517-534.
Rouse, W.B. 1988. Adaptive aiding for human/computer control. *Human Factors, 30*:431-438.
Shafer, G. 1996. *The art of causal conjecture*. Cambridge, MA: The MIT Press.
Sheridan, T.B. 1988. Task allocation and supervisory control. In *Handbook of human-computer interaction,* ed. M. Helander. NY: North-Holland.
Sheridan, T. 1992. *Telerobotics, automation, and supervisory control*. Cambridge, MA: MIT Press.
Sperling, G. 1978. The attention operating characteristic: Examples from visual search. *Science, 202*:315-318.
Stillwell, W.G., Seaver, D.A., and Schwartz, J.P. 1981. *Expert estimation of human error probabilities in nuclear power plant operations: A review of probability assessment and scaling* (Report No. NUREG/CR-2255). Washington, DC: Nuclear Regulatory Commission.
Thornton, R.C., Kaempf, G.L., Zeller, J.L. and McAnulty, D.M. 1991. *An evaluation of crew coordination and performance during a simulated UH-60 helicopter mission.* Fort Rucker, AL: U. S. Army Research Institute Aviation Research and Development Activity.
Toulmin, S. 1958. *The uses of argument*. Cambridge, UK: Cambridge University Press.
Tversky, A., and Kahneman, D. 1973. Availability: A heuristic for judging frequency and probability. *Cognitive Psychology, 4*:207-232.
Tversky, A., and Kahneman, D. 1974. Judgment under uncertainty: Heuristics and biases. *Science, 185*:1124-1131.
Wason, S.R. 1960. On the failure to eliminate hypotheses in a conceptual task. *Quarterly Journal of Experimental Psychology, 12*:129-140.
Wellens, A.R. 1993. Group situation awareness and distributed decision making: From military to civilian applications. In *Individual and group decision making,* ed. N. J. Castellan, Jr. Hillsdale, NJ: Erlbaum.
Wickens, C.D. 1984. Processing resources in attention. In *Varieties of attention,* ed. R. Parasuraman and R. Davies, 63-101. NY: Academic Press.
Zuboff, S. 1988. *In the age of the smart machine: The future of work and power*. NY: Basic Books.

APPENDIX A: THE APT-R MODEL

In this appendix, which can be read in conjunction with Chapter 3, we will describe the APT-R model more generally and more formally. The model draws upon, and adapts, the decision theoretic measure of value of information, as presented, for example, in Raiffa & Schlaifer (1961, pp. 79-92) and explored in Cohen & Freeling (1981).

The adaptations of the value of information measure that we present are designed to address two shortcomings of traditional models. First, classical decision theory requires that all options be specified in advance. In the real-world, by contrast, decision makers are more likely to focus on one option at a time, moving on to evaluate another option only when problems are found (Simon, 1957; Klein, 1993). In addition, part of the problem solving process may include finding or inventing options that were unknown when the process began. Secondly, classical decision theory requires that the outcomes of information collection options be specified in advance, when those outcomes can affect subsequent decisions. In real-world settings, by contrast, it is sometimes impossible to anticipate all the possible things that might be learned by new observation or new thinking.

APT-R is not intended to model real-world behavior. However, it is intended to provide a benchmark against which real-world behavior can be evaluated.[18] Thus, key features of APT-R include devices for dealing with both incremental discovery and evaluation of options, and situations in which decision makers can discover genuinely new (i.e., unanticipated) information. Tools from the classical model, when adapted in this way, can shed light on the concept of trust, just as trust enables us to create a more naturalistic version of the classical model.

In sum, this model is not intended as a description of the internal cognitive processes of decision aid users. Rather, its usefulness is as a tool: (1) for clarifying the concept of trust and its relation to reliance decisions, (2) for generating practice scenarios in which decision aid users can make reliance decisions under varying conditions of decision aid accuracy, and (3) for developing benchmarks and corresponding feedback for user performance.

A Reliance Decision without Verification

We will first consider a simple, classical model of the user's response to an aid's recommendation when there is no possibility of further verifying it. The components of this decision are options, uncertain states of the world, and outcomes.

Options. Suppose the decision aid has made a recommendation r, e.g., regarding the selection of a battle position or the classification of a contact. We should think of r as the entire output of the aid with regard to a particular problem. For example, r may include more than one aid recommendation (e.g., classifications of a contact, or potential battle positions), ranked in order of confidence.

$a_1(r)\ldots a_m(r)$ are a set of options from which the user could in principle choose for final dispensation of r (e.g., accept the top-ranked aid recommendation, modify the top-ranked recommendation in any of various ways, reject the top-ranked recommendation and accept the second ranked alternative, and so on). Note that the a_i refer to user interactions with the aid, such as "accepting," "rejecting," or "modifying." The actual action corresponding to $a_i(r)$ is a function of r, as indicated by the notation, since, for example, accepting a recommendation to engage is quite different from accepting a recommendation not to engage. We will represent the actual actions that might be considered by the user as a function a(r) of the aid recommendation, even though the aid may in some cases have little real influence on the option actually adopted, for example, when the user ends up locating a battle position on a paper map. The number of potential options $a_1(r)\ldots a_m(r)$ can be very large, e.g., including all the possible battle positions within a given region that a user might end of accepting.

For simplicity, we can suppress the reference to r, and designate the user's interaction options as $a_1\ldots a_m$, in contexts where the aid's recommendation is already known at the time of the user's decision.

Uncertain states of the world and probabilities. The $s_1\ldots s_p$ are an exhaustive and mutually exclusive set of states of the world about which the user is uncertain, but which determine the degree of success of whatever action a_k the user selects. In Chapter 2, we looked at event trees containing variables (such as rotorwash and angle of attack) which influence the appropriateness of a battle position recommendation. The $s_1\ldots s_p$ are simply a convenient way of talking about all possible combinations of values for such variables (i.e., their Cartesian product). For example, s might specify the status of a battle position with respect to angle of attack, rotorwash, and all other features that determine whether the user's selected battle position is acceptable. For a target identification aid, s might specify the true classification of the contact. Uncertainty about these states can be represented by means of probabilities, as discussed in Chapter 2.

[18] See Cohen (1993) for a argument that prescriptive models should fit qualitative aspects of the behavior they are intended to guide.

Outcomes and utilities. Outcomes are simply combinations of actions and states of the world, e.g., accepting a battle position that has rotorwash and a rear angle of attack, rejecting a battle position that has no rotorwash and a frontal angle of attack, etc. The user's preferences for these outcomes are represented by a utility or value function on every combination of action and state of the world: $\mathbf{u}_f(a,s)$. The final utility function, $\mathbf{u}_f(a,s)$, is the utility of the outcome produced by action a under conditions s (for example, the value of accepting the recommended battle position when it has particular features of rotorwash and angle of attack).

Figure 31 depicts the components of a reliance decision, when verification is not possible. Probabilities are assessed for each state of the world s, and utilities for each combination of action and state of the world (a,s).

Given these components, a measure of the desirability of each option can be calculated. To do so, we start at the terminal nodes on the far right of the tree, and work toward the left, calculating the expected utility of each node in a process called *averaging out and folding back* (Raiffa, 1968). The expected utility of chance nodes (depicted by circles) is represented by an expectation (or averaging) operator. For example, the final expected utility of selecting action a_j is:

$$\mathbf{u_f}(a_j) = \mathbf{E}_s\, \mathbf{u_f}(a_j,s)$$
$$= \sum_i \mathbf{p}(s_i)\, \mathbf{u_f}(a_j,s)$$

To calculate expected utility at a chance node, we multiply the probability on each branch leading from the node by the utility or expected utility to which it leads, and sum across all the branches. In this case, we get the probability-weighted average of the final utility function $\mathbf{u_f}(a_j,s)$, over possible states of the world s, given that final action a_j will be chosen (and assuming that s is a discrete and finite variable). The expected utility of decisions (depicted by square nodes) is represented by the **max** operator. For example, the expected utility of the decision above is:

$$\mathbf{u}_f = \mathbf{max}_a\, \mathbf{E}_s\, \mathbf{u_f}(a,s)$$

This represents the expected utility of whatever action a has the largest expected final utility, $\mathbf{u_f}(a,s)$, over possible values of s.

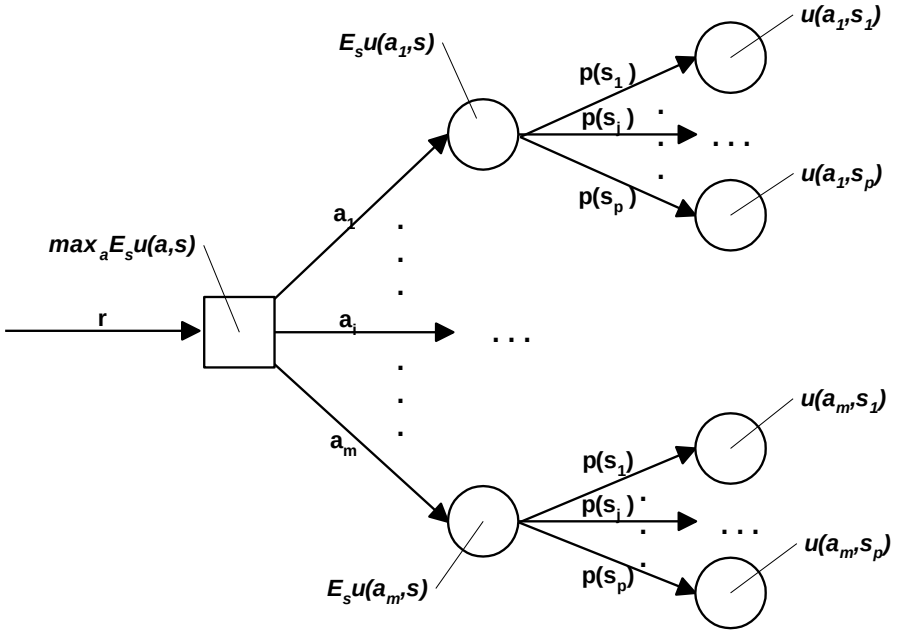

Figure 31. Classical model of a reliance decision without the possibility of verification. Ellipses indicate branches that are not shown.

Varieties of Trust

Thus far, our model of the reliance decision does not refer to *trust*. By a simple reframing of the model, however, the central role of trust can be clarified. This reframing involves introducing a new *partition* (an exhaustive and mutually exclusive classification) of the outcomes s, based on what action is optimal in each. Partitions of this kind will also be useful when we turn to the possibility of verifying an aid recommendation.

For any situation s, there is a subset of the options $a_1 \ldots a_m$ which would be acceptable in s[19]. Let us define the *acceptability class* S_j of an action a_j as the set of situations s in which a_j would be acceptable after uncertainty about

[19] In some situations or tasks, the only acceptable option may be the best or optimal option. In other cases, e.g., selecting a battle position, multiple options may clear an acceptability threshold and it is not particularly important

s has been resolved. In other words, if a particular option a_j (e.g., a battle position) would be acceptable to a decision maker once the decision maker has full knowledge of s (e.g., rotorwash, angles of attack, and so on), then s belongs to S_j. We can use the concept of an acceptability class to define several useful partitions of the situations s.[20]

Trust in the decision aid recommendation. We will let $a_1(r)$ designate (without loss of generality) *acceptance* of the aid's recommendation r. To assess trust in the decision aid, we do not need to consider the full range of options $a_1(r)\ldots a_m(r)$ in detail; we need only consider $a_1(r)$, and the user's expectation that it will be successful. This is the probability that the future state of affairs s, when it becomes known, will belong to the acceptability class S_1 (in which a_1 is acceptable). In a decision regarding acceptance of a_1, the user need only consider the chances that the future situation will be in S_1 or $\neg S_1$.

We will define a partition S_a of the states of the world s that is just fine enough to determine correct acceptance or rejection of the aid recommendation. $S_a = \{S_1, \neg S_1\}$, where S_1 is the set of all situations s in which a_1 is acceptable, and $\neg S_1$ is the complementary set of situations, in which a_1 is not acceptable. Thus, $\mathbf{p}(S_1) = 1 - \mathbf{p}(\neg S_1)$ = *trust in the decision aid recommendation* in the narrow sense defined in Chapter 2. In terms of decision trees, this corresponds to the sum of the probabilities of all paths that lead from the node where a_1 has been selected, to a successful outcome.

Figure 32 depicts the calculation of the expected utility of a_1 based on the partition S_a. From the previous section, we have:

$$\mathbf{u}_f(a_1) = \mathbf{E}_s\, \mathbf{u}_f(a_1,s)$$
$$= \mathbf{E}_{Sa}\, \mathbf{E}_{s\,|\,Sa}\, u(a_1,s)$$
$$= \mathbf{p}(S_1)\, \mathbf{E}_{s\,|\,S1}\, \mathbf{u_f}(a_1,s) + \mathbf{p}(\neg S_1)\, \mathbf{E}_{s\,|\,\neg S1}\, \mathbf{u_f}(a_1,s).$$

We can define a new utility function on the partition S_a by taking the average utility in the finer partition based on s:

$$\mathbf{u_f}(a_1, S_a) = \mathbf{E}_{s\,|\,Sa}\, \mathbf{u_f}(a_1,s).$$

Thus,

$$\mathbf{u}_f(a_1) = \mathbf{p}(S_1)\, \mathbf{u_f}(a_1, S_1) + \mathbf{p}(\neg S_1)\, \mathbf{u_f}(a_1, \neg S_1) = \mathbf{E}_{Sa}\, \mathbf{u}(a_1,S_a)$$

$\mathbf{u}_f(a_1)$ is trust in the broader sense introduced by Chapter 3, in which outcomes are weighted by their degree of desirability, or utility. The probability of success is weighted by the expected, or average, utility of successful situations, $\mathbf{E}_{s\,|\,S1}\, \mathbf{u_f}(a_1,s)$; and the probability of failure is weighted by the expected utility of unsuccessful situations, $\mathbf{E}_{s\,|\,\neg S1}\, \mathbf{u_f}(a_1,s)$.

Trust in the decision aid-user interaction. We can use the same approach to define trust in the overall user-decision aid interaction, after the user has chosen one of the options $a_1(r)\ldots a_m(r)$. We will designate a' as the element of $a_1(r)\ldots a_m(r)$ that the user selects, whatever it happens to be. The user's preferred alternative reflects the results of user-decision aid interaction, rather than the conclusions of either the user or the aid alone.

---

to identify the "best." The determination of an acceptability threshold depends on the goals and risks in the overall task or mission, but we assume that the threshold is reasonable, in the sense that there is always at least one acceptable action in a situation. The converse, however, does not hold: There may well be actions which are acceptable in no situation.

[20] Note that the acceptability classes themselves, $S_1\ldots S_j\ldots S_m$, are not necessarily a partition of the situations s. Since more than one action might be acceptable in a given situation s, s can belong to more than one of the $S_1\ldots S_j\ldots S_m$, which are therefore not mutually exclusive. However, if only the best option in a situation is acceptable (and if ties between equally good actions are broken, e.g., by tossing a coin), then each situation s will belong to one and only one acceptability class, and $S = \{S_i \text{ for all } i\}$ will be a partition of the situations.

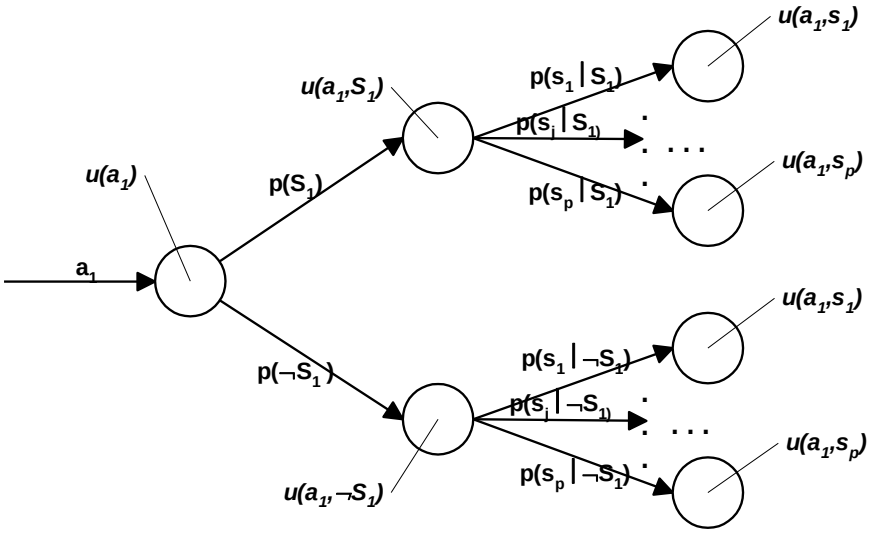

Figure 32. Trust in the decision aid recommendation a_1, based on a partition of situations s into a class of situations S_1 in which a_1 would be acceptable and a class $\neg S_1$ in which it is not.

To calculate the chance of success of a′, we identify another partition, S_I, of the same situations $s_1 \ldots s_p$, which is based on the acceptability of a'. $S_I = \{S', \neg S'\}$ where S' is the set of all situations s in which a' is acceptable, and $\neg S'$ is the complementary set of situations, in which a' is not acceptable. The expected utility of a' is:

$$\mathbf{u}_f(a') = \mathbf{E}_{SI}\, \mathbf{E}_{s \mid SI}\, u(a',s)$$
$$= \mathbf{p}(S')\, \mathbf{E}_{s \mid S'}\, \mathbf{u}_f(a',s) + \mathbf{p}(\neg S')\, \mathbf{E}_{s \mid \neg S'}\, \mathbf{u}_f(a',s)$$
$$= \mathbf{p}(S')\, \mathbf{u}_f(a', S_1) + \mathbf{p}(\neg S')\, \mathbf{u}_f(a', \neg S') = \mathbf{E}_{SI}\, u(a', S_I)$$

p(S') is trust in the result of user-aid interaction, i.e., the chance that the currently preferred option will lead to an acceptable outcome. $\mathbf{u}_f(a') = \mathbf{E}_{SI}\, \mathbf{E}_{s \mid SI}\, u(a',s)$ is trust in the broader sense, in which probabilities of success and failure are weighted by their respective average utilities.

Upper bound on trust in user-aid interaction. There is a limit to how much it is reasonable to expect from user-decision aid interaction. We can derive this limit by looking at the result of optimal user decision making. We will designate a* as the interactive response to r that an ideal user would adopt; i.e., a* is the element of $a_1(r) \ldots a_m(r)$ that has the highest expected utility:

$$\mathbf{u}_f(a^*) = \mathbf{max}_a\, \mathbf{E}_s\, \mathbf{u}_f(a,s)$$

a* is the best action the user could adopt given uncertainty about the true state of affairs s. But a* is not necessarily an appropriate option for every situation s. A user with advance knowledge that a particular s will turn out to be the case might in fact not choose a*. Trust in a* is the chance that a* will turn out to be acceptable in the situation that actually occurs. To capture this, we introduce a third partition S_B of the situations $s_1 \ldots s_p$, which is based on the acceptability of a*. Let $S_b = \{S^*, \neg S^*\}$ where S* is the set of all situations s in which a* turns out to be acceptable, and $\neg S^*$ is the complementary set of situations. The expected utility of a* is:

$$\mathbf{u}_f(a^*) = \mathbf{E}_{Sb} \mathbf{E}_{s \mid Sb}\, u(a^*,s)$$
$$= \mathbf{p}(S^*)\, \mathbf{E}_{s \mid Sb}\, \mathbf{u}_f(a^*,s) + \mathbf{p}(\neg S^*)\, \mathbf{E}_{s \mid \neg Sb}\, \mathbf{u}_f(a^*,s)$$
$$= \mathbf{p}(S^*)\, \mathbf{u}_f(a^*, S^*) + (1 - \mathbf{p}(S^*))\, \mathbf{u}_f(a^*, \neg S^*) = \mathbf{E}_{Sb}\, u(a^*, S^*)$$

$\mathbf{u}_f(a^*)$ provides an upper bound on the utility-weighted degree of trust the user can reasonably have in any option.

Binary decisions. If the aid's conclusion is binary (e.g., identification as friend or foe), the evaluation of user interaction options in terms of trust takes an interesting form. In that case, there are only two options: a_1 and a_2. One of these, a_1, must be acceptance of the aid recommendation, and the other, a_2, rejection. The user should accept the aid recommendation if:

$$\mathbf{u}_f(a_1) > \mathbf{u}_f(a_2)\text{, or}$$
$$\mathbf{p}(S_1)\, \mathbf{E}_{s \mid S1}\, \mathbf{u}_f(a_1,s) + \mathbf{p}(S_2)\, \mathbf{E}_{s \mid S2}\, \mathbf{u}_f(a_1,s) > \mathbf{p}(S_1)\, \mathbf{E}_{s \mid S1}\, \mathbf{u}_f(a_1,s) + \mathbf{p}(S_2)\, \mathbf{E}_{s \mid S2}\, \mathbf{u}_f(a_1,s).$$

We can simplify (and clarify) this by substituting as follows:

$$\textit{expected cost of incorrectly rejecting aid recommendation} = \mathbf{E}_{s \mid S1}\, [\, \mathbf{u}_f(a_1,s) - \mathbf{u}_f(a_2,s)\,] = \mathbf{c}_r$$
$$\textit{expected cost of incorrectly accepting aid recommendation} = \mathbf{E}_{s \mid S2}\, [\, \mathbf{u}_f(a_2,s) - \mathbf{u}_f(a_1,s)\,] = \mathbf{c}_a.$$

Since $\mathbf{p}(S_2) = 1 - \mathbf{p}(S_1)$, it follows that the user should accept the aid recommendation (i.e., take action a_1) if:

$$\textit{trust in the aid} = \mathbf{p}(S_1) > \mathbf{c}_a / (\mathbf{c}_a + \mathbf{c}_r)\,.$$

Conversely, the user should reject the aid's recommendation (i.e., take action a_2) if:

$$\textit{trust in aid} = \mathbf{p}(S_1) < \mathbf{c}_a / (\mathbf{c}_a + \mathbf{c}_r) .$$

A Model of the Verification Decision

When verification is possible, the user need not choose immediately among the $a_1 \ldots a_m$, but may choose to gather additional information first in order to make a more informed decision. The verification decision introduces three new components: verification options, verification outcomes, and verification costs. (See Figure 33.)

Verification options. In the verification decision the user faces a choice among options: v_0, representing the decision not to verify the recommendation, and $v_1, \ldots v_n$, representing the available options for making observations, requesting information, performing additional analyses, exploring modifications of the current option, identifying or creating new options, and so on.

Verification outcomes. If users choose a verification option $v_i \neq v_0$, they will observe one member of the mutually exclusive and exhaustive set, $z_{i,1} \ldots z_{i,q}$, of possible observational outcomes of the chosen verification process, v_i. Each $z_{i,}$ is thus a different random variable, representing the possible outcomes of a different verification option. (For example, let v_i = looking at angle of attack for a recommended battle position, $z_{i,1}$= frontal angle of attack, and $z_{i,2}$ = rear or flanking angle of attack.) If users do not verify (v_0), they collect no further information, which we will represent by the dummy observation z_0, and they immediately choose among the $a_1 \ldots a_m$ for the current aid recommendation r.

It is notoriously hard to deal with the discovery or creation of new options in classical decision theory. Recall that in our model of the reliance decision without verification, we stipulated that $a_1(r) \ldots a_m(r)$ include all possible user responses to the aid's recommendation. We retain that stipulation now, but drop the usual implication that the decision maker is aware of all those options at the time of the decision. We will model the discovery or creation of new options by generalizing the concept of *verification*, which we define as a process of collecting information that can change the expected utility of options.

Our approach rests on two extensions of the classical value of information model: (1) Subsets of the possible options $a_1(r) \ldots a_m(r)$ of which the user is not aware at any given time are represented as having expected utility of zero. (2) Verification options $v_1, \ldots v_n$ include collection of information about the existence of new options. Such verification options might include, for example, looking for promising battle positions in a particular area of the map, modifying a plan, or trying to think of new hypotheses about the intent of a contact to explain its behavior. Such verification options, if adopted, can lead to observations (e.g., finding a good battle position, improving a plan, or thinking of a likely intent) that raise the utility of an option in the set $a_1(r) \ldots a_m(r)$ from zero to some positive quantity. As a result, when the user chooses among the $a_1(r) \ldots a_m(r)$ after verification, there may be effectively more choices available than if the user had chosen among the $a_1(r) \ldots a_m(r)$ before verification.

Verification costs. When verification is possible, a complete path through the decision tree consists of the user selecting a verification option, v_i , observing some outcome $z_{i,j}$, selecting a final action a_k, and experiencing an outcome determined by s_h. The user's preferences are represented by a utility or value function on such paths through the decision tree: $\mathbf{u}(v,z,a,s)$. A key assumption of this analysis is that this utility function can be decomposed into two additive parts, $\mathbf{u}(v,z,a,s) = \mathbf{u}_v(v,z) + \mathbf{u}_f(a,s)$, corresponding to the cost of the verification process and the value of the final action, respectively (Raiffa & Schlaifer, 1961, pp. 79-81). The final utility function, $\mathbf{u}_f(a,s)$, is the same function that we saw earlier in our analysis of a reliance decision without verification. The verification utility function, $\mathbf{u}_v(v,z)$, reflects the cost in time and risk of performing the observations or analyses represented by verification process v. We will assume that this cost or risk, $\mathbf{u}_v(v,z)$, is independent of the actual outcome z of the observations. Because $\mathbf{u}_v(v,z)$ will typically be a negative quantity, it is convenient to express it in terms of cost instead of utility. Combining the latter two considerations, we get:

$$\mathbf{c}(v_i) = - \mathbf{u}_v(v_i, z_i)$$

The costs of different verification options can vary considerably. For example, the cost of not verifying, $\mathbf{c}(v_0)$, is zero. However, the cost of making the observations required to identify new battle positions from a paper map might be quite high.

Figure 33 shows how these components of the verification decision appear in a decision tree. Note that the decision not to verify (v_0) leads to the same probabilities and expected utilities as when verification was not possible at all (Figure 31).

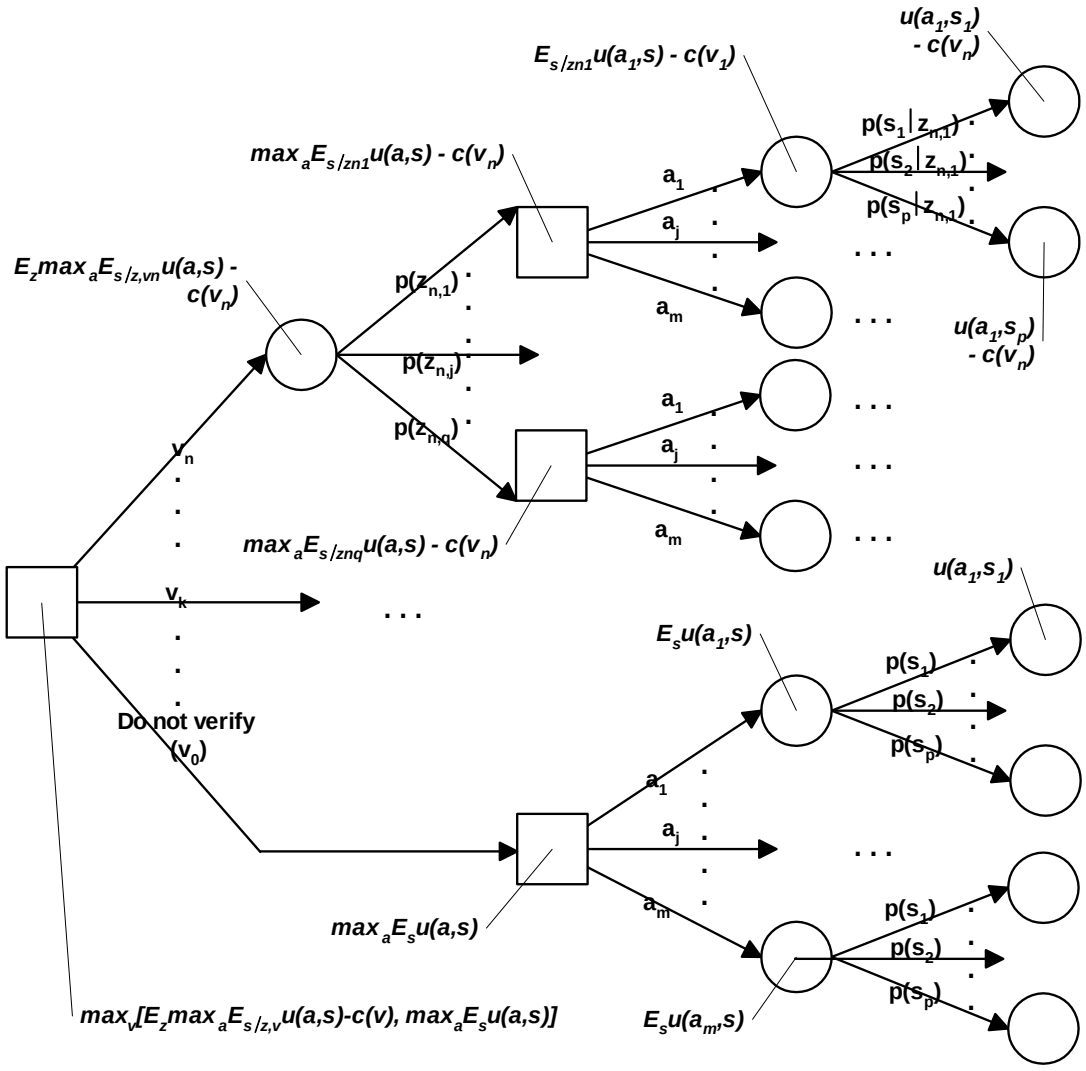

Figure 33. Model of a verification decision.

Expected utilities can be calculated for each node in this tree by the same process of averaging out and folding back that we illustrated earlier. We apply the expectation (averaging) operator at the chance nodes representing uncertain states of the world just as in the earlier example, except that probabilities are now conditional on the verification option and verification outcome on the path leading to that node. For example, the final expected utility of adopting verification process v_1, observing $z_{1,1}$, and then selecting action a_2 is:

$$\mathbf{EU}_f(v_1,z_{1,1},a_2) = \mathbf{E}_{s\,|\,v1,z1,1}\,\mathbf{u}_f(a_2,s)$$
$$= \sum_i \mathbf{p}(s_i\,|\,v_1,z_{1,1})\,\mathbf{u}_f(a_2,s)$$

This is the probability-weighted average of the final utility function $\mathbf{u}_f(a_2,s)$, over possible states of the world s, *given* that the verification option v_1 will be selected, observation $z_{1,1}$ will be observed, and final action a_2 will be chosen (again assuming that s is a discrete and finite). After performing verification process v_1 and observing $z_{1,1}$, the expected utility of the final decision, is:

$$\mathbf{EU}_f(v_1,z_{1,1}) = \mathbf{max}_a\,\mathbf{E}_{s\,|\,v1,z1,1}\,\mathbf{u}_f(a,s)$$

This represents the expected utility of whatever action a has the largest final utility, $\mathbf{u}_f(a,s)$, averaged over possible values of s, *given* $z_{1,1}$ and v_1.

We use the expectation operator at chance nodes for verification outcomes just as we did for uncertain states of the world. Thus, the final expected utility of a verification option, say v_1, is the average, over its possible observational outcomes z_1, of the expected value of the final action:

$$\mathbf{EU}_f(v_1) = \mathbf{E}_{z1\,|\,v1}\,\mathbf{max}_a\,\mathbf{E}_{s\,|\,v1,z1}\,\mathbf{u}_f(a,s)$$

In the special case of the option not to verify, v_0, no information is actually collected; thus, the expected utility of v_0 is simply the result of selecting the action a that maximizes utility averaged over values of s:

$$\mathbf{EU}_f(v_0) = \mathbf{E}_{z0\,|\,v0}\ \mathbf{max}_a\ \mathbf{E}_{s\,|\,v01,z0}\ \mathbf{u_f}(a,s)$$
$$= \mathbf{max}_a\ \mathbf{E}_s\ \mathbf{u_f}(a,s)$$

Value of Information. The value of information (**VOI$_f$**) for any verification option, say v_i, is simply the difference between the final expected utility of v_i and the final expected utility of immediately selecting an action (v_0):

$$\mathbf{VOI_f}(v_i) = \mathbf{EU}_f(v_i) - \mathbf{EU}_f(v_0)$$
$$= \mathbf{E}_{zi\,|\,vi}\ \mathbf{max}_a\ \mathbf{E}_{s\,|\,vi,zi}\ \mathbf{u_f}(a,s) - \mathbf{max}_a\ \mathbf{E}_s\ \mathbf{u_f}(a,s)$$

It is reasonable to assume the verification process itself does not affect the probabilities of situations, i.e., the probability of a situation s given that z_i is true is the same whether or not v_i was performed and z_i was in fact observed. We can thus drop the conditioning on v_i. In addition, $\mathbf{E}_s\ \mathbf{u_f}(a,s)$ is equivalent to a weighted average of its values over possible values of z_i (whether or not z_i is actually observed by the user). Thus,

$$\mathbf{VOI_f}(v_i) = \mathbf{E}_{zi}\ \mathbf{max}_a\ \mathbf{E}_{s\,|\,zi}\ \mathbf{u_f}(a,s) - \mathbf{max}_a\ \mathbf{E}_{zi}\ \mathbf{E}_{s\,|\,zi}\ \mathbf{u_f}(a,s)$$

It is now clear that verifying and not verifying differ only in the relative position of expectation operators **E** and maximization operators **max**. In decision tree terms, they differ in whether or not in the user will collect formation about z_i before deciding among the final options $a_1 \ldots a_m$. **VOI$_f$** is equivalent to the expected *cost of errors*, i.e., the utility that would be lost due to not having that information. Equivalently, value of information is the extra utility expected by waiting to observe the value of z_i before acting rather than acting immediately on the currently favored action.

Earlier we defined a' as the user's currently preferred action. In the present context, this means that a' is whatever action the decision maker would choose if unable to verify; in other words, a' is the member of $a_1 \ldots a_m$ (e.g., acceptance, modification, or rejection of the aid conclusion) that maximizes average utility over the possible values of z_i and s *without* knowledge of z_i. Thus:

$$\mathbf{VOI_f}(v_i) = \mathbf{E}_{zi}\ \mathbf{max}_a\ \mathbf{E}_{s\,|\,,zi}\ \mathbf{u_f}(a,s) - \mathbf{E}_{zi}\ \mathbf{E}_{s\,|\,zi}\ \mathbf{u_f}(a',s)$$

Simplifying, we get:

$$\mathbf{VOI_f}(v_i) = \mathbf{E}_{zi}\ [\ \mathbf{max}_a\ \mathbf{E}_{s\,|\,zi}\ \mathbf{u_f}(a,s) - \mathbf{E}_{s\,|\,zi}\ \mathbf{u_f}(a',s)\ ]$$

This formulation is of special interest, and is the source of the verbal statement in Chapter 3. It represents the average over the possible observations z_i, of the difference between the utility expected from the options that would be selected given knowledge of z_i and the utility expected from the currently preferred option a′. Information collection is valueless if there is no observation that would cause decision makers to change their preference among options,

Thus far, we have neglected the cost $\mathbf{c}(v_i)$ of verification process v_i. Users choose a verification option by maximizing expected utility (or trust) with respect to the *total* utility function $\mathbf{u}(v_i,z,a,s)$. Given the additive decomposition of that function, the chosen option will be the v_i such that:

$$\mathbf{max}_{vi}\ \mathbf{EU}(v_i) = \mathbf{max}_{vi}\ \mathbf{E}_{zi}\ \mathbf{max}_a\ \mathbf{E}_{s\,|\,zi}\ \mathbf{u}(v,z,a,s)$$
$$= \mathbf{max}_{vi}\ \mathbf{E}_{zı}\ \mathbf{max}_a\ \mathbf{E}_{s\,|\,zi}\ [\ \mathbf{u_f}(a,s) - \mathbf{c}(v_i)\ ]$$
$$= \mathbf{max}_{vi}\ [\ \mathbf{E}_{zı}\ \mathbf{max}_a\ \mathbf{E}_{s\,|\,zi}\ \mathbf{u_f}(a,s) - \mathbf{c}(v_i)\ ]$$
$$= \mathbf{max}_{vi}\ [\ \mathbf{VOI_f}(v_i) - \mathbf{c}(v_i) + k\ ]$$

where k is the expected utility of the currently preferred option a′. The best verification option is simply the v_i for which the value of information $\mathbf{VOI_f}(v_i)$ exceeds the cost $\mathbf{c}(v_i)$ by the largest amount.

Trust and Dynamic Constraints on Verification.

It is sometimes said that decision theoretic models require that the possible outcomes of each option be specified in advance. This is not strictly true. Standard models require that the expected utility of each option be assessed. One way to do this is to assess probabilities and utilities for each outcome of each action. However, it is also possible to specify no outcomes at all, and to assess the expected utility of each action directly. The latter strategy would be analogous to the Phase 2 estimates of trust illustrated in Figure 8, which are directly associated with specific conditions (external backing) rather than being based on anticipation of future events (internal backing). Another way of saying this is that the event trees that follow decisions can be as detailed or as aggregated as one likes. Unfortunately, decision theoretic models lose this flexibility when actions involve the collection of information that may affect *future decisions*. As we saw in the previous section, modeling future decisions requires maximization of expected utility conditional on the observations (z) that might result from the information collection actions (v). Thus, the possible outcomes of information collection (z) must be specified in advance. The event trees that intervene between the verification decision and the final choice among the $a_1 \ldots a_m$ must be complete.

This requirement is troublesome for a variety of reasons outlined in Chapter 3: e.g., the results of visual observation, conflict resolution, or accumulation of evidence are often unpredictable. In addition, this requirement undercuts our

own device of representing the discovery or creation of new options as a possible outcome of verification. If verification outcomes must be known in advance, then the options that they uncover must also be known in advance. In this section, we describe a way to recover the flexibility of decision models with respect to specifying the outcomes of verification. The solution is to give up estimating the exact expected utility of a verification action (v), and to substitute constraints on the conditions under which verification may be appropriate.

Dynamic constraints on verification can be derived by starting with the definition of expected value of information (**VOI_f**) for verification process v_i in the last section:

$$\mathbf{VOI_f}(v_i) = \mathbf{E}_{zi} [\mathbf{max}_a \mathbf{E}_{s|zi} \mathbf{u_f}(a,s) - \mathbf{E}_{s|zi} \mathbf{u_f}(a',s)]$$

Let us assume that the decision aid user has done the best possible job identifying an appropriate option, short of resolving uncertainty about s. Thus, the currently preferred option a′ is the same as the option a* that maximizes $\mathbf{E}_s$ $\mathbf{u_f}(a,s) = \mathbf{E}_{zi}$ $\mathbf{E}_{s|zi}$ $\mathbf{u_f}(a,s)$.

The maximum that **VOI_f** can take is the expected value of *perfect* information (**$VOPI_f$**), when the observation z provides full foreknowledge of the situation s. In that case, we can substitute s for z and simplify:

$$\mathbf{VOPI_f}(v_i) = \mathbf{E}_s [\mathbf{max}_a \mathbf{u_f}(a,s) - \mathbf{u_f}(a',s)]$$

This maximum value can be achieved, however, even if the decision maker does not learn the precise identity of the situation s. All that is required is that the decision maker learn enough to choose the correct action a. To capture this requirement, we will define a generalization of **$VOPI_f$** that we will call the expected value of *partial (but) perfect* information (**$VOPPI_f$**). In this case, the observation z may, but does not necessarily, provide full knowledge of which situation s is the case. At the least, however, z provides sufficient knowledge of which final action a should be adopted. In other words, z will tell the user to *correctly* accept a′ or else *correctly* select from among the remaining a. Recall that we defined the subset of the situations $s_1 \ldots s_p$ in which action a_j would be acceptable as the *acceptability class* S_j. In the case of partial but perfect information, z produces *perfect* knowledge of the acceptability class to which the true situation belongs.

It will be necessary to convert the set of acceptability classes S = {S_i for all actions a_i} into a partition, i.e., to ensure that every s belongs to one and only one S_i. We can do this most simply, and consistently with our goal of constructing an upper bound, by requiring that an acceptable option be the best available (see footnote 20). Thus, a situation s belongs to the acceptability class S_i if and only if a_i has the highest expected utility in s. By assumption, z will provide perfect knowledge that some member of S is the case. Thus, we can substitute S for z in the first equation in this section:

$$\mathbf{VOPPI_f}(v_i) = \mathbf{E}_S [\mathbf{max}_a \mathbf{E}_{s|S} \mathbf{u_f}(a,s) - \mathbf{E}_{s|S} \mathbf{u_f}(a',s)]$$

Since knowledge of S exhausts the possibilities for changing the final decision a, **$VOPPI_f$** is quantitatively identical to the value of perfect information (**$VOPI_f$**). But **$VOPPI_f$** allows for the possibility that z will not discriminate further among the situations s, even though these further distinctions affect utility, $\mathbf{u_f}(a,s)$. For example, suppose there are two possible actions: to engage a contact (a_1) or not to engage a contact (a_2). Thus, there are two acceptability classes S_1 and S_2 in which each action is appropriate, respectively. Suppose further that there are three relevant situations, each of which belongs to one or the other of the two acceptability classes, as shown in the following table:

This situation	**belongs to this acceptability class**	**associated with this action**
s_1 = enemy tank	S_1	a_1= engage
s_2 = friendly truck	S_2	a_2= do not engage
s_3 = enemy truck	S_2	a_2= do not engage

Suppose a target identification aid recommends engagement of the contact. Acceptance of the recommendation to engage is correct if the contact is in fact an enemy tank, and incorrect if it is a friendly truck *or* an enemy truck. The distinction between a friendly truck (s_2) and enemy truck (s_3) is irrelevant for the engagement decision, since both belong to the same acceptability class S_2, and the aid does not need to discriminate them (i.e., the value of that information is zero with respect to this decision). However, the distinction between destroying an enemy and destroying a friendly has an enormous impact on the cost of a mistaken engagement. As we shall see, the relative balance of enemy and friendly vehicles can also have an enormous impact on the level of trust that a user requires before accepting the aid's recommendation to engage.

As another example, suppose a battle position has a 70% chance of success, but an unfavorable angle of attack would shift the likelihood of success down to 40%. The battle position is unacceptable in either case, yet the cost of mistakenly adopting this battle position will be quite different in the two cases, i.e., a 30% swing in chance of success.

Information has value only to the extent that it can change decisions. Thus, our definition of the value of partial but perfect information can be simplified if we incorporate the partition S_I that we introduced earlier based on user interaction with the aid. S′ is the set of all situations s in which a′ remains preferred after resolving uncertainty about s. The more likely that S′ is the case, the less value there is in collecting information. The more likely that ¬S′ is the case (i.e., a′ is not appropriate), the more important it is to resolve the uncertainty.

The relationship between this partition, S_I , and the partition S consisting of all acceptability classes, is straightforward. The preferred option based on user-aid interaction, a′, must be identical to some single option a_j, and thus S′ is identical to some specific acceptance class S_j On the other hand, if a' is not appropriate, different options may be preferred depending on the situation, and so ¬S' may include more than one acceptance class in S.

We now extend the definition of $\mathbf{VOPPI_f}$ based on these relationships:

$$\mathbf{VOPPI_f}(v_i) = \mathbf{E}_S [\mathbf{max}_a \mathbf{E}_{s|S} \mathbf{u_f}(a,s) - \mathbf{E}_{s|S} \mathbf{u_f}(a',s)]$$

$$= \mathbf{p}(S') [\mathbf{max}_a \mathbf{E}_{s|S'} \mathbf{u_f}(a,s) - \mathbf{E}_{s|S'} \mathbf{u_f}(a',s)] + \mathbf{p}(\neg S') \mathbf{E}_{S|\neg S'} [\mathbf{max}_a \mathbf{E}_{s|S} \mathbf{u_f}(a,s) - \mathbf{E}_{s|S} \mathbf{u_f}(a',s)]$$

When S' is the case, the preferred option a' maximizes expected utility, and the first addend is zero The value of verification therefore focuses on the complement of S':

$$\mathbf{VOPPI_f}(v_i) = \mathbf{p}(\neg S') \mathbf{E}_{S|\neg S'} [\mathbf{max}_a \mathbf{E}_{s|S} \mathbf{u_f}(a,s) - \mathbf{E}_{s|S} \mathbf{u_f}(a',s)]$$

$\mathbf{VOPPI_f}$ represents the best-case outcome of verification: sufficient knowledge to determine the best action. Verification process v_i cannot be appropriate if $\mathbf{VOPPI_f}(v_i) < \mathbf{c}(v_i)$, i.e., if the best that verification can accomplish is worth less than its cost. It follows algebraically that verification process v_i is inappropriate, if:

$$\mathbf{p}(\neg S') < \mathbf{c}(v_i) / \mathbf{E}_{S|\neg S'} [\mathbf{max}_a \mathbf{E}_{s|S} \mathbf{u_f}(a,s) - \mathbf{E}_{s|S} \mathbf{u_f}(a',s)]$$

or equivalently, since $\mathbf{p}(\neg S') = 1 - \mathbf{p}(S')$,

$$\mathbf{p}(S') > 1 - \mathbf{c}(v_i) / \mathbf{E}_{S|\neg S'} [\mathbf{max}_a \mathbf{E}_{s|S} \mathbf{u_f}(a,s) - \mathbf{E}_{s|S} \mathbf{u_f}(a',s)]$$

$\mathbf{p}(S')$ is the chance of a situation in which a' is successful. Thus, $\mathbf{p}(S')$ is trust in the overall user-aid interaction, which resulted in the user's current preference for a'. The constraint itself reflects the ratio of costs $\mathbf{c}(v_i)$ of a particular verification strategy v_i to its expected benefits *given* that the currently preferred option is wrong, $\mathbf{E}_{S|\neg S'} [\mathbf{max}_a \mathbf{E}_{s|S} \mathbf{u_f}(a,s) - \mathbf{E}_{s|S} \mathbf{u_f}(a',s)]$. If trust is high enough so that the constraint is satisfied for all verification options, the currently preferred action should be accepted without verification.

We can simplify the form of the constraint considerably by framing it explicitly in terms of the costs and conditional benefits (i.e., the "upside") of a verification strategy. Such a strategy can be appropriate only if:

trust < 1 – costs / conditional benefits

An interesting feature of this constraint is its implication for the differential impact of changes in cost and changes in conditional benefit. Suppose we are trying to decide which of several verification strategies to use. Figure 16 in Chapter 3 shows that no strategy is worth considering unless its conditional benefits exceed its costs (ratio > 1.0). As long as this is the case, however, upgrading one's verification strategy by increasing costs in exchange for greater potential benefits may not be worthwhile. In most cases, increases in cost will have a far greater impact on the upper bound for trust than the equivalent increases in benefit. Hence, a lower level of trust will be required to warrant the "upgraded" verification strategy, versus remaining with a lower cost, lower benefit verification approach. To examine the relative sensitivity of the upper bound to costs and benefits, we take the derivative of the upper bound with respect to cost and with respect to conditional benefit (*cbenefit*), and then look at the ratio of the derivatives. The derivative of the upper bound with respect to cost is -1/cbenefit. The derivative of the upper bound with respect to cbenefit is cost/cbenefit^2 . Thus, the ratio of the rate of change with cost to the rate of change with cbenefit is cbenefit / cost. For example, if the conditional benefit of a verification strategy is twice its costs, an increase of cost must be offset by twice as large an increase in conditional benefit, to keep the threshold for verification at the same level of trust.

Using constraints to compare verification options. Verification is defined as an information collection process that can change the expected utility of options. As a result, both the benefits and costs of a verification strategy depend on the number of options whose expected utility it can alter. As shown in Table 16, we can subdivide the options available to the user, and classify verification strategies according to the scope of their influence on these subsets of options.

The constraint on verification derived in the previous section can be used to compare these different verification options with one another. Constraints can be defined for each type of strategy based on its costs and benefits, as shown in Table 17.

Each constraint provides an upper bound on trust for the corresponding class of strategies. If trust in the currently preferred option, $\mathbf{p}(S')$, exceeds the constraint for a given strategy, it is not worthwhile to pursue that strategy, although some other verification strategy (e.g., with a lower cost and/or greater potential benefit) may be justified. These constraints are illustrated in Figure 18 in Chapter 3.

Table 16. Three classes of verification strategies.

This type of verification strategy	affects expected utility of these options		with this average cost
V_1	$A_1 = \{a_{1,1}(r) \ldots a_{1,m}(r)\}$	*the currently known options (e.g., listed explicitly by the aid)*	$\mathbf{c}(V_1)$
V_2	$A_2 = \{a_{1,1}(r) \ldots a_{1,m}(r), a_{2,1}(r) \ldots a_{2,n}(r)\}$	*+ options that can be obtained by modifications of the known options*	$\mathbf{c}(V_2)$
V_3	$A_3 = \{a_{1,1}(r) \ldots a_{1,m}(r), a_{2,1}(r) \ldots a_{2,n}(r), a_{3,1}(r) \ldots a_{3,p}(r)\}$	*+ new options that can be discovered or created*	$\mathbf{c}(V_3)$

Table 17. Constraints on each type of verification strategy.

This verification class	is appropriate only if this constraint is true:
V_1	$\mathbf{p}(S') < 1 - \mathbf{c}(V_1) / \mathbf{E}_{S \mid \neg S'} [\mathbf{max}_{a \in A1} \mathbf{E}_{s \mid S} \mathbf{u_f}(a,s) - \mathbf{E}_{s \mid S} \mathbf{u_f}(a',s)]$
V_2	$\mathbf{p}(S') < 1 - \mathbf{c}(V_2) / \mathbf{E}_{S \mid \neg S'} [\mathbf{max}_{a \in A2} \mathbf{E}_{s \mid S} \mathbf{u_f}(a,s) - \mathbf{E}_{s \mid S} \mathbf{u_f}(a',s)]$
V_3	$\mathbf{p}(S') < 1 - \mathbf{c}(V_3) / \mathbf{E}_{S \mid \neg S'} [\mathbf{max}_{a \in A3} \mathbf{E}_{s \mid S} \mathbf{u_f}(a,s) - \mathbf{E}_{s \mid S} \mathbf{u_f}(a',s)]$

Binary conclusions. If the aid's conclusion is binary (e.g., identification as friend or foe), we can derive the somewhat simpler constraints stated in Chapter 3. In that case, there are only two final options: a' (which is currently preferred) and a". Moreover, there are only two acceptance classes, viz., S', implying that a' is appropriate, and S", implying that a" is appropriate. Finally, there is only one basic verification strategy, V_1, which involves seeking evidence bearing on both a' and a". We can now simplify the inequalities above. Verification is inappropriate if:

$$\mathbf{p}(\neg S') < \mathbf{c}(V_1) / \mathbf{E}_{s \mid S''} [\mathbf{u_f}(a'',s) - \mathbf{u_f}(a',s)] \text{ , and, equivalently,}$$

$$\mathbf{p}(S') > 1 - \mathbf{c}(V_1) / \mathbf{E}_{s \mid S''} [\mathbf{u_f}(a'',s) - \mathbf{u_f}(a',s)]$$

If a' (the option that is preferred prior to verification) happens to be *acceptance* of the aid's conclusion (i.e., $a' = a_1$), then $\mathbf{p}(S') = p(S_1)$ is equivalent to *trust in the decision aid alone*. Then the second inequality above can be rewritten as:

$$\mathbf{p}(S_1) > 1 - \mathbf{c}(V_1) / \mathbf{E}_{s \mid S2} [\mathbf{u_f}(a_2,s) - \mathbf{u_f}(a_1,s)]$$

If this constraint is satisfied, the aid's conclusion should be accepted (and its negation rejected) without verification. On the other hand, if the currently favored option a' happens to be *rejection* of the aid's conclusion (i.e., $a' = a_2$), then $\mathbf{p}(\neg S') = 1 - \mathbf{p}(S') = p(S_1)$ is trust in the aid, and the first inequality above can be rewritten as:

$$\mathbf{p}(S_1) < \mathbf{c}(V_1) / \mathbf{E}_{s \mid S1} [\mathbf{u_f}(a_1,s) - \mathbf{u_f}(a_2,s)]$$

If this constraint is satisfied, the aid's conclusion should be rejected (and its negation accepted) without verification. Combining the two constraints, verification is permissible only if:

$$1 - \mathbf{c}(V_1) / \mathbf{E}_{s \mid S2} [\mathbf{u_f}(a_2,s) - \mathbf{u_f}(a_1,s)] > \mathbf{p}(S_1) > \mathbf{c}(V_1) / \mathbf{E}_{s \mid S1} [\mathbf{u_f}(a_1,s) - \mathbf{u_f}(a_2,s)].$$

This is the source of the upper and lower bounds in Figure 17. Note that the costs of different kinds of errors appear in the denominators of each constraint (viz., the cost of performing a_1 when a_2 would be appropriate in the denominator on the left, and the cost of performing a_2 when a_1 would be appropriate on the right), and that these costs depend on the distribution of situations s within S_1 and S_2. Thus, for example, the relative proportion of friendly vehicles and enemy trucks among non-targets will influence the threshold for acceptance of an identification friend-or-foe conclusion. As the ratio of friendly vehicles increases, the amount of trust required for engagement increases.

We can find the conditions at which the upper and lower bound meet, eliminating the possibility of further verification. Earlier, we defined the following abbreviations:

$$\textit{cost of incorrectly rejecting aid's conclusion} = \mathbf{E}_{s \mid S1} [\mathbf{u_f}(a_1,s) - \mathbf{u_f}(a_2,s)] = \mathbf{c}_r$$

$$\textit{cost of incorrectly accepting aid's conclusion} = \mathbf{E}_{s \mid S2} [\mathbf{u_f}(a_2,s) - \mathbf{u_f}(a_1,s)] = \mathbf{c}_a.$$

When the two constraints for binary verification are equal, we have:

$$1 - \mathbf{c}(V_1) / \mathbf{c}_a = \mathbf{c}(V_1) / \mathbf{c}_r ,$$

It follows that verification is no longer appropriate when:

$$\mathbf{c}(V_1) = \mathbf{c}_r \mathbf{c}_a / (\mathbf{c}_r + \mathbf{c}_a)$$

At this value of cost, the two constraints will have converged on the following value:

$$1 - \mathbf{c}(V_1) / \mathbf{c}_a = \mathbf{c}(V_1) / \mathbf{c}_r = \mathbf{c}_a / (\mathbf{c}_a + \mathbf{c}_r)$$

Not surprisingly, this is the trust criterion that would determine which action should be taken if verification were not possible. As we saw earlier, the expected utility of a_1 is greater than the expected utility of a_2 if and only if:

$$\mathbf{p}(S_1) > \mathbf{c}_a / (\mathbf{c}_a + \mathbf{c}_r)$$

When the upper and lower bound converge, the user's decision is equivalent to a decision without the possibility of verification.

APPENDIX B. SOURCES OF EVIDENCE FOR PROBLEMS

Table 8 in Chapter 4 lists 20 potential problems in user interaction with decision aids. The following is a sampling of the empirical evidence for each of these problems.

P1: Riley, V., Lyall, B., & Wiener, E. (1993). *Analytic methods for flight-deck automation design and evaluation. Phase two report: Pilot use of automation*. (Technical Report). Minneapolis, MN: Honeywell Technology Center.

P2: Riley, V. (1996). Operator reliance on automation: Theory and data. In R. Parasuraman, & M. Mouloua (Eds.), *Automation and human performance: Theory and applications*. Hillsdale, NJ: Erlbaum.

P3: Scerbo, M. W., Greenwald, C. Q., & Sawin, D. A. (1992). Vigilance: It's boring, it's difficult, and I can't do anything about it. In *Proceedings of the Human Factors and Ergonomics Society, 36,* (pp. 1508-1511). Santa Monica, CA: Human Factors and Ergonomics Society.

P4: Kirlik, A. (1993). Modeling strategic behavior in human-automation interaction: Why an "aid" can (and should) go unused. *Human Factors, 35*, 221-242.

P5: Riley, V. (1996). Operator reliance on automation: Theory and data. In R. Parasuraman, & M. Mouloua (Eds.), *Automation and human performance: Theory and applications*. Hillsdale, NJ: Erlbaum.

P6: Lee, J. D., & Moray, N. (1994). Trust, self-confidence, and operators' adaptation to automation. *International Journal of Human-Computer Studies, 40*, 153-184.

P7: Riley, V. (1996). Operator reliance on automation: Theory and data. In R. Parasuraman, & M. Mouloua (Eds.), *Automation and human performance: Theory and applications*. Hillsdale, NJ: Erlbaum.

P8: Roth, E. M., Bennett, K. B., & Woods, D.D. (1987). Human interaction with an "intelligent" machine. *International Journal of Man-Machine Studies, 31*, 517-534.

P9: Sarter, N., & Woods, D. D. (1994). Pilot interaction with cockpit automation II: An experimental study of pilots' model and awareness of the flight management system. *The International Journal of Aviation Psychology, 4*, 1-28.

P10: Parasuraman, R., Molloy, R., & Singh, I.L. (1993). Performance consequences of automation-induced "complacency." *The International Journal of Aviation Psychology, 3,* 1-23.

P12: Sarter, N., & Woods, D. D. (1994). Pilot interaction with cockpit automation II: An experimental study of pilots' model and awareness of the flight management system. *The International Journal of Aviation Psychology, 4*, 1-28.

P13: Cohen, M. S. (1993). The naturalistic basis of decision biases. In G.A. Klein (Ed.*), Decision making in action*. Norwood, NJ: Ablex.

Mosier, K., & Skitka, L. J. (1996). Human decision makers and automated decision aids: Made for each other? In R. Parasuraman, & M. Mouloua (Eds.*) Automation and human performance: Theory and applications*. Mahwah, NJ: Erlbaum.

P14: Tversky, A., & Kahneman, D. (1984). Judgment under uncertainty: Heuristics and biases. *Science, 185,* 1124-1131.

P15: Rogers, W. H., Schutte, P. C., & Latorella, K.A . (1996). Fault management in aviation systems. In R. Parasuraman, & M. Mouloua (Eds.), *Automation and human performance: Theory and applications*. Hillsdale, NJ: Erlbaum.

P16: Getty, D. J., Swets, J. A., Pickett, R. M., & Gounthier, D. (1995). System operator response to warnings of danger: A laboratory investigation of the effects of the predictive value of a warning on human response time. *Journal of Experimental Psychology: Applied, 1*, 19-33.

Parasuraman, R., Hancock, P. A., & Olofinboba, O. (1997). Alarm effectiveness in driver-centered collision-warning systems. *Ergonomics, 40,* 390-399.

P17: Casner, S. (1994). Understanding the determinants of problem-solving behavior in a complex environment. *Human Factors, 36*, 580-596.

P18: Wickens, C.D. (1992). *Engineering psychology*. New York: Harper.

P19: Funk, K., Lyall, B., & Riley, V. (1995*). Perceived human factors problems of flightdeck automation. (Phase 1 Final Report).* Corvallis, OR: Oregon State University.

P20: Foushee, H. C., & Helmreich, R.L. (1988). Group interaction and flight crew performance. In E.L. Wiener & D. C. Nagel (Eds.), *Human factors in aviation.* (pp. 189-227). San Diego: Academic Press.

APPENDIX C. ILLUSTRATIVE RPA TRAINING PACKAGE

The following is the demonstration training package described in Chapter 5 for the Combat Battle Position Recommendation aid.

APPENDIX D. TRUST AND ASSOCIATE BEHAVIOR IN RPA

Introduction

This section discusses how some subtle, but very important RPA behavior issues are determined by the heavy employment of branching *constraints* contained in the CIE and Task Network knowledge representations. The CIE uses a set of constraints, depending on the granularity of the distinction between one sibling node to another in the PGG, to make the best determination of crew intentions based on observed actions and current context. Context is determined by dynamic variables that are monitored and evaluated by CIE as it attempts to identify crew intent. Constraint reasoning is a key coordination link between the crew and the RPA's context-sensitive, proactive aiding capability. Additionally, it provides a rich set of customization parameters with the ability to tailor behavior trigger thresholds, etc. Training sessions held with advanced RPA crews could potentially lead to the crew altering context parameters dynamically to customize RPA so that crews would develop a better functional model of RPA behavior: Trust is achieved through understanding the RPA's predictable behavior within context.

In the following sections, the topic of crew action interpretation is covered as a consequence of discussing intent interpretation within context. A discussion of the constraint set problem leads into constraint representation in the CIE (for the production of training materials), followed by an example vignette from which materials for RPA interaction training could be derived.

General Constraint Discussion

Definition of a Constraint

CIE employs constraints to determine context. A simple constraint is typically represented as a conditional relationship between a set of variables of interest. For example, a constraint template might be:

Continue to attack if <number of missiles>

is greater than SAFE_RETURN_NUMBER

where:

<number of missiles> is a *variable* representing the current number of missiles left on board and SAFE_RETURN_NUMBER is a *constant* .

Instances of the <number of missiles> variable could be 0, 1 or some other realistic number, for example, that is returned by an access to the ownship state vector. SAFE_RETURN_NUMBER is a *constant*. that is predetermined at knowledge base load time, and can be customized. "Using this constraint in context" means that when the system attempts to evaluate the constraint template, it will grab different values of the variable type specified in the template. For example, the instance of the constraint would be true if <number of missiles> available is 4 and SAFE_RETURN_NUMBER was 1. Likewise, if <number of missiles> equaled 0, the constraint would fail.

Purpose of Constraints in CIE and Constraint Satisfaction

CIE uses the above approach for constraint representation to introduce context into the intent interpretation process, so that it can notify other RPA functions of the crew's desires without explicitly asking. The process of constraint satisfaction is described mathematically in the following way.

The system is given a set of variables, a finite and discrete domain for each variable, and a set of constraints. Each constraint is defined over some subset of the original set of variables and limits the combination of values that the variables can take (refer to the above example).

Each time CIE is activated with a new action to explain, constraints are instanced and evaluated to determine if the action is a fit, ***in context***., as CIE conducts bottom up interpretation of the current action. If the context is incorrect, the constraint fails, and CIE keeps looking for an explanation until one is found. If none are found, the action is unexplained.

Should a suitable explanation be found (e.g., he is reacting to actions on contact (AOC) due to new threat, etc.), the context is preserved and propagated to all interested RPA consumers. In this way, the system has an idea of the conditions under which the RPA/crew team is behaving.

The evaluation and maintenance of constraining relationships on RPA behavior is an important synchronization mechanism in the RPA. When even subtle changes in the current context are determined, the system reevaluates the active constraints to determine if its behavior is still appropriate for the changed situation.

Use of Context Constraints in RPA Training and Operations

The PGG link constraints provide a rich area on which to build trust. The constraint relationships and context model-extracted variable values define the deep context under which crew actions are interpreted. In this section, we analyze the constructs of the CIE knowledge bases (i.e., plan-goal graph, constraint sets and parameters) that affect the establishment and maintenance of trust between the crew and RPA. An example vignette is proposed where the

effect of how intent is determined using constraint-based reasoning, within dynamic context, directly affects this relationship.
Through training, the crew could understand how context is considered in determining intent. Using this operational knowledge the crew can be more certain of what the RPA will do in replicable situations, "cue" certain RPA behaviors by acting in a certain manner, and learn how to adjust constraints and context values to tailor very subtle decisions that the RPA is making.

Constraints and Parameters RPA

Recall that the RPA is an "electronic crewmember" that interprets what goals crew is attempting to address and provide assistance in executing plans (tasks) towards achieving those goals. A subtle, but nonetheless important aspect of constraint processing is that the combination of the knowledge-engineered PGG and link constraints allow the RPA to be smart about context determination. It is therefore empowered to make subtle distinctions as to why the crew executed certain actions *in a particular context.*
Constraints put the conditions on the CIE's ability to link observed crew actions to potential explanations while traversing the plan-goal graph. Constraint relationships exist as "link constraints," where a link between a child and parent node is made context-conditional via constraint sets. An example PGG subtree is shown in Figure 34 with link constraints enumerated below.

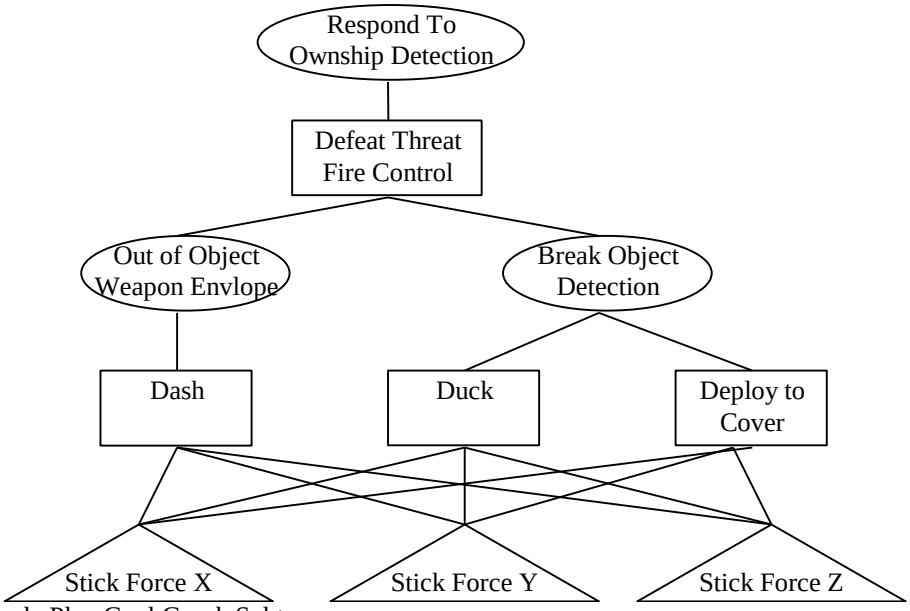

Figure 34. Example Plan Goal Graph Subtree.
Note that all stick actions are connected to all parents. Context determines the interpretation. The context information determining link traversal for this subtree are :

- Presence of an Imminent Threat
- Threat detecting ownship
- Ownship following BRASSCRAF recommendation
- Change in heading

Thread constraints are the most fine-grained constraints, disambiguating the action down to a detailed plan for achieving their intentions.. Although somewhat cryptic, context access method names, such as "getEntityRangeFromOwnship," actually go out to the context model to retrieve a parameter value determined by the algorithm in the parameter calculation. In the example "getEntityRangeFromOwnship", the method returns the actual distance from ownship to the current world entity (e.g., air defense vehicle). It is compared with a constant that is tailorable by either the knowledge engineer or other capable party.
Each of the threads are delineated from others by solving the constraint sets. Note that the actions being evaluated are all the same -- stick control actions. They are being interpreted differently based on establishment of context and solution of the constraint sets.
The constraints by thread for linking stick actions to potential explanation of intent for this example are:
Respond To Ownship Detection - Dash

Thread -

- GOAL Respond To Ownship Detection
- PLAN Defeat Threat Fire Control
- GOAL Out Of Object Weapon Envelope
- PLAN Dash
- ACTION Stick Force

Constraining Values -

Respond To Ownship Detection ⇔ Defeat Threat Fire Control

NONE

Defeat Threat Fire Control ⇔ Out Of Object Weapon Envelope

◊ detection range >= getEntityRangeFromOwnship

Out Of Object Weapon Envelope ⇔ Dash

◊ getNumberOfImminentThreats >= 1
◊ getTrueAirspeed NOT WITHIN 50 of getPlannedAirspeed
◊ getTrueAirspeed >= getPlannedAirspeed
◊ offRoute <= 100
◊ getYawAcceleration NOT WITHIN 5 of 0
◊ getTrueHeading WITHIN 3 getBRASSCRAFRecommendedHeading

Dash ⇔ Stick Force

NONE

Respond To Ownship Detection - Duck

Thread -

- GOAL Respond To Ownship Detection
- PLAN Defeat Threat Fire Control
- GOAL Break Object Detection
- PLAN Duck
- ACTION Stick Force

Constraining Values -

Respond To Ownship Detection ⇔ Defeat Threat Fire Control

NONE

Defeat Threat Fire Control ⇔ Break Object Detection

◊ detection range >= getEntityRangeFromOwnship

Break Object Detection ⇔ Duck

◊ offRoute <= 100
◊ getAGLAltitude <= getPlannedAltitude
◊ getAGLAltitude WITHIN 10 of getManeuverAltitude
◊ getNumImmThreats >= 1
◊ getYawAcceleration NOT WITHIN 5 of 0
◊ getTrueHeading WITHIN 3 getBRASSCRAFHeading

Duck ⇔ Stick Force

NONE

Respond To Ownship Detection - Deploy To Cover

Thread -

- GOAL Respond To Ownship Detection
- PLAN Defeat Threat Fire Control
- GOAL Break Object Detection
- PLAN Deploy To Cover

- ACTION Stick Force

Constraining Values -

Respond To Ownship Detection ⇔ Defeat Threat Fire Control

NONE

Defeat Threat Fire Control ⇔ Break Object Detection

◊ detection range >= getEntityRangeFromOwnship

Break Object Detection ⇔ Deploy To Cover

◊ offRoute >= 100
◊ getNumImmThreats >= 1
◊ getAGLAltitude <= getPlannedAltitude
◊ getAGLAltitude NOT WITHIN 10 of getPlannedAltitude
◊ getTrueHeading WITHIN 3 getBRASSCRAFHeading

Deploy To Cover ⇔ Stick Force

◊ getYawAcceleration NOT WITHIN 5 of 0
◊ getTrueHeading WITHIN 3 getBRASSCRAFHeading

PGG Link Constraint Factors Affecting Trust

There are three factors contained in the link constraint sets above that affect the crew's ability to understand CIE / RPA behavior and therefore trust:

- Constants (e.g., 5 knots),
- Context Variables (e.g., getEntityRangeFromOwnship),
- Combination of constraints on links,

where Constants would be thresholds, variances, and limits; Context Variables represent the same utility as constants, only the values are dynamically bound by context source (e.g., current aircraft flight parameters, etc.); and the Combination of constraints by PGG link that would help tailor action interpretation in context.

In essence, the constraints represent a "deal" between crew and the RPA that helps understand crew actions in different environment settings. They could be used by the crew or designers of CIE and CIM to adjust to, for example:

- flying style (loose vs. tight),
- desired level of interaction with RPA in different situations (more experienced),
- risk calibration by crew by mission type,
- subsystem (e.g., weapons) customization.

Vignette: Auto-Recalibration of Task Network by CIE during Actions on Contact to Visual Threat

The RPA has the ability to engage in mixed initiative, specifically during the actions on contact situation,. Typically, a great deal of training is given to crews about how to handle actions on contact situations. The RPA has, at its roots, the ability to deliver the crew from this situation using point automation for specific support (e.g., BA automatically slewing sensors to identified targets, etc.), UNLESS the RPA is unaware of the threat. Unfortunately, in this case, the RPA *has no idea* that there is a threat present.

In our vignette, the crew is responding to a single troop (or small group) of hostiles with hand-held weapons (e.g., SA-7). It unfolds as follows:

While flying along the planned reconnaissance route, the pilot spots a single troop on a hilltop bringing an SA-7 to bear on ownship. The pilot engages in classic actions on contact behavior, wherein he:

-- decouples the autopilot,
-- rapidly descends looking for cover,
-- arms the gun and slaves the sight to HMD, and
-- brings the gun to bear

BUT, there is no known "imminent" threat to BA and therefore RPA cannot determine AOC in the standard way. So, the normal constraint network fails (see PER-20, Section 4.0). What will fire is a similar version of this PGG branch called "Respond to VISUAL threat." The interesting point is that the constraint set is the same, except "imminent threat existing" is not part of the constraint set.

This subtle case, highlighted during the actions on contact threads, initially is concerned with the identification of the crew's intentions to engage in AOC. Once AOC is determined, the interpretation of the crew's actions, through constraint processing, will lead CIE to conclude that he is deploying to cover in responding to a visual object.
The constraint sets are used for two purposes. Initially, the CIE attempts to identify the behavior exhibited by the crew during the actions on contact situation. The parameters identified in the constraint set determine the context thresholds and limits used to explain the crew's behavior. Customization of constraint parameters can be made by a crew or knowledge engineer to help determine if the crew is going to SvP indicated cover, own cover, or using a different tactic. In this way the system can interpret subtle behavior differences.
Further, an experienced crew could execute tasks in specific ways to cue a desired, predictable RPA behavior.