N94- 35063

Vista Goes Online: Decision-Analytic Systems for Real-Time Decision-Making in Mission Control

Matthew Barry
Propulsion Systems Section
NASA/Johnson Space Center DF63
Houston, TX 77058

Eric Horvitz (PI), Corinne Ruokangas, Sampath Srinivas
Information and Decision Sciences
Rockwell International Science Center
444 High Street
Palo Alto, CA 94301

Abstract

The Vista project has centered on the use of decision-theoretic approaches for managing the display of critical information relevant to real-time operations decisions. The Vista-I project originally developed a prototype of these approaches for managing flight control displays in the Space Shuttle Mission Control Center (MCC). The follow-on Vista-II project integrated these approaches in a workstation program which currently is being certified for use in the MCC. To our knowledge, this will be the first application of automated decision-theoretic reasoning techniques for real-time spacecraft operations.

We shall describe the development and capabilities of the Vista-II system, and provide an overview of the use of decision-theoretic reasoning techniques to the problems of managing the complexity of flight controller displays. We discuss the relevance of the Vista techniques within the MCC decision-making environment, focusing on the problems of detecting and diagnosing spacecraft electromechanical subsystem component failures with limited information, and the problem of determining what control actions should be taken in high-stakes, time-critical situations in response to a diagnosis performed under uncertainty. Finally, we shall outline our current research directions for follow-on projects.

1 Introduction

The Vista project is a collaborative research and development effort between the Palo Alto Laboratory of the Rockwell Science Center, the Rockwell Space Operations Company, and NASA/Johnson Space Center to develop techniques for reducing the cognitive load on operators responsible for monitoring and controlling complex physical systems. In particular, the project has centered on the use of decision-theoretic approaches for generating diagnostic assistance and for directing computer programs to display the most relevant information in a decision context.

Last year, we developed and demonstrated a prototype Vista-I decision-support and display-management system for Space Shuttle Orbital Maneuvering System (OMS) burn monitoring and control activities. This prototype system provides propulsion system flight controllers with diagnostic decision support by reasoning under uncertainty about alternative problems, and by prioritizing them according to probability and criticality [1].

This Vista-I prototype stimulated efforts to continue this work by extending the reasoning models and porting the techniques to MCC-class workstations, culminating with certification of the software for mission operations. To accomplish these efforts, the Vista team this year developed the Vista-II system. This system improves the Vista-I uncertainty models, supplements them with utility models, and captures the prototyped display-management features and techniques within an X-windows-based workstation program connected to the MCC telemetry data streams. The resulting program currently is undergoing final development and verification and validation testing prior to certification.

2 Description

The proper management of uncertainty in decision-making is critically important in high-risk operations endeavors like manned space flight. The Space Shuttle OMS performs many critical maneuvers (commonly called *burns*) during every mission, including orbit insertion and deorbit, rendezvous target phasing and orbital plane adjustments, deployed-satellite and collision-avoidance separations, and contingency propellant dumps. Therefore, it is vitally important that correct OMS diagnoses and operations decisions be made promptly when subsystem faults occur during these maneuvers.

The set of possible faults is known *a priori*, as are the valid responses to any combination of these failures. Since the OMS subsystem is well-transduced, the fault detection and diagnosis tasks are rather straightforward for an experienced flight controller; a less-experienced flight controller, however, may have a bit more uncertainty about fault signatures and correct response actions. However, any flight controller faces significantly more difficult decision-making tasks when a prior failure of the spacecraft instrumentation or data processing subsystems has rendered many of the primary OMS sensors inoperative. Our program-embedded uncertainty models handle this often-encountered situation by using whatever information is available in the current situation, including secondary sensors and prior probabilities. Moreover, prior problems within the OMS subsystem may increase the difficulty of diagnosing multiple faults; the uncertainty models handle these situations in an elegant manner because they calculate the probability distribution over all faults.

In Vista applications, we use uncertainty models to calculate the probability distributions over the set of possible faults based on observed sensor data. We use these probability distributions in conjunction with utility models to determine which course of action to recommend. Both of these models affect the automated selection of adaptive displays which the program provides to flight controllers for making the final diagnosis and response decisions. Sections 2.1 and 2.2 describe the uncertainty and utility models, respectively, and section 3 describes the displays and display-management techniques we've built into the Vista-II system.

2.1 Uncertainty Model

Automated reasoning systems often require representations of uncertainty about the world. These models often employ Bayesian inferencing techniques to calculate conditional probabilities over a collection of hypotheses given some evidence. They are especially applicable to fault detection and diagnosis problem domains in which multiple faults may occur or in which only a limited amount of evidence is available. Vista systems employ these models within larger decision-theoretic models representing uncertainty and utility in decision-making processes. In Vista systems we apply these uncertainty models to the usual problems of fault detection and diagnosis, but we also apply them to the problem of automatically controlling the presentation of information to the user given uncertainty about the world.

Vista systems use *belief network* models to calculate the probability distributions over a set of possible faults for the OMS rocket engines and their associated propellant distribution systems and sensors. Belief networks are computational models which represent probabilistic influences among observations (evidence) and possible explanations for these observations (conclusions or diagnoses).[1] In the OMS burn monitoring and control program specifically, we use belief networks to represent the probabilistic influences among telemetered readings from OMS pressure, temperature, quantity, and valve position sensors against a collection of possible faults or explanations which best describe these observations. Figure 1 depicts a compact representation of the OMS burn network.

Each belief network node contains conditional probabilities based on the conditional probabilities of its ancestors. We enter observations from the world as certain evidence in certain leaf nodes. The inference engine propagates this evidence, using Bayes' Rule, to all of the other nodes in the network. Extracting the resulting values of features within designated fault nodes we obtain the conditional probability distribution for given exhaustive set of faults. The program uses this fault probability distribution to update and manage displays and as input into the utility model.

2.2 Utility Model

For automated decisions about the best action to take under uncertainty, it is important to employ a representation of the value of alternative outcomes. Having access to the values of alternative outcomes allows for the selection of fault-response actions that have the highest expected utility. In the Vista-II system we employ a *utility model* to calculate the value of alternative outcomes based on the fault probability distribution. We display the distribution of these values over all of the alternative actions and assume that the flight controller will select the action with the maximum expected utility. Section 3 describes these displays.

The Vista-II utility model determines the value of alternative outcomes by calculating the scalar product of the fault probability distribution vector with an action-specific, utility-weighting parameter vector. We have experimented with various sets of weighting parameters. The set currently in place reflects a single-attribute model which describes the "right response" or "gut feeling" gleaned from experienced flight controllers. Essentially, these parameters reflect the utility of selecting action A in response to each possible fault F. We have also constructed more specific multi-attribute utility models which can provide the weighting vector elements by performing a linear combination of decision attributes. These decision attributes include measures such as the importance of achieving maneuver targets (based on criticality), the risk of damage to spacecraft subsystem components, the per-

[1] These belief networks, sometimes referred to as *causal probability networks*, are special forms of more general *influence diagrams* [2].

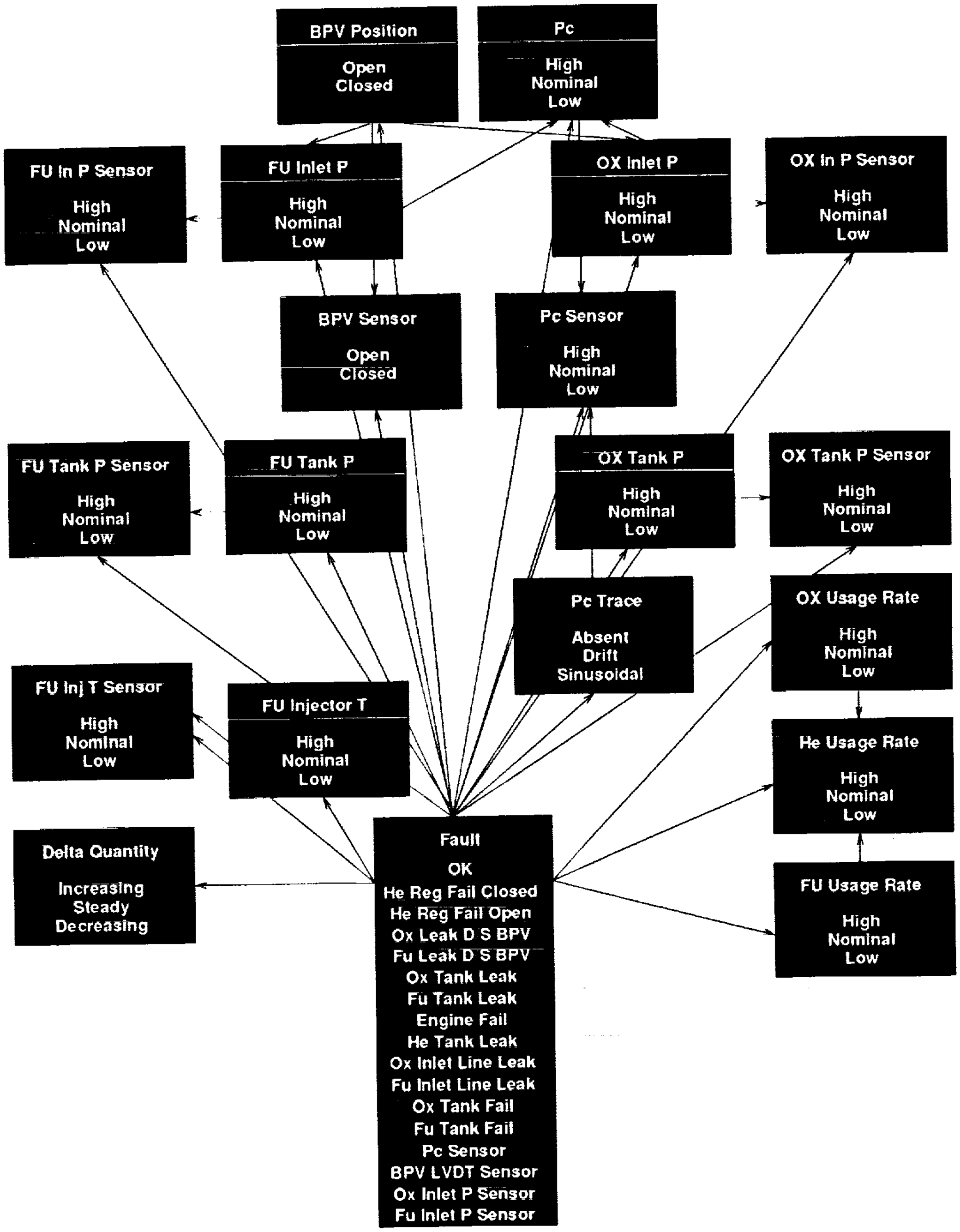


Figure 1: The belief network for OMS burn monitoring. Arcs represent probabilistic influences between the nodes. Grayed titles denote evidence nodes.

formance capabilities available from backup systems, and the potential impact to mission objectives. These multi-attribute utility models will be particularly important in distributed decision-making applications because they provide a way to account for disparate degrees of contribution from independent subsystems toward common decision attributes.

The model we have implemented in the Vista-II system provides the flight controller with a utility value distribution over four possible actions. These actions correspond to doing nothing ("continue"), terminating the burn ("stop"), or selecting a backup burn configuration ("engine-fail downmode" and "propellant-fail downmode"). Since the expected utility of executing these actions in response to a fault is context-dependent, the utility model employs a different set of weighting parameters for each user-selected context. Section 3 describes the mechanism by which the user can select the context. As the fault probability distribution changes according to the uncertainty model, the utility model changes the distribution over these possible actions and the program shows this distribution on the displays.

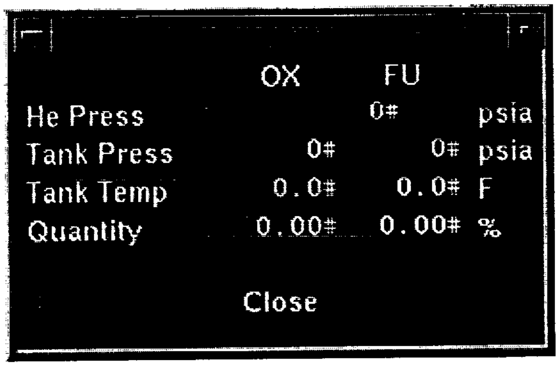
3 Implementation

The Vista-II application has been realized in a working program on MCC-class workstations. These workstations run the Unix operating system and the X-Windows System, and use the OSF/Motif window manager. The OMS burn monitoring program was written in the C language using the OSF/Motif programming style and widget set and the Hugin API inference engine for the belief networks.[2] Owing to our lack of access to a commercial product performing utility modeling, we have coded the utility models by hand. In this section we describe some of the implementation techniques, display-management philosophies, and design details found in this program.

First, since the two OMS engines are functionally identical, but provide unique sets of sensor values, we use a copy of the belief network for each engine and change the engine-specific sensor value evidence nodes according to the appropriate sensor names and locations. The belief network developer assigns to each fault given in the "fault" node an associated "group" name, which we use to collect related faults into named groups in order to summarize these faults on a smaller display. The OMS burn monitoring program loads these two belief networks at run time. Once loaded and initialized, the program constructs some of the Vista displays automatically based on the contents of the designated "fault" node in the network. The program then cyclically gathers telemetered sensor values, translates analog values into qualitative values (such as `low`, `nominal`, or `high`), then installs these qualitative values as certain evidence in the sensor nodes. If the value of any evidence node has changed since the last data cycle, the program runs the belief network inference engine to compute the probability distributions over all of the possible values in each of the other nodes. The program then uses these new probability distributions to update and select the appropriate displays.

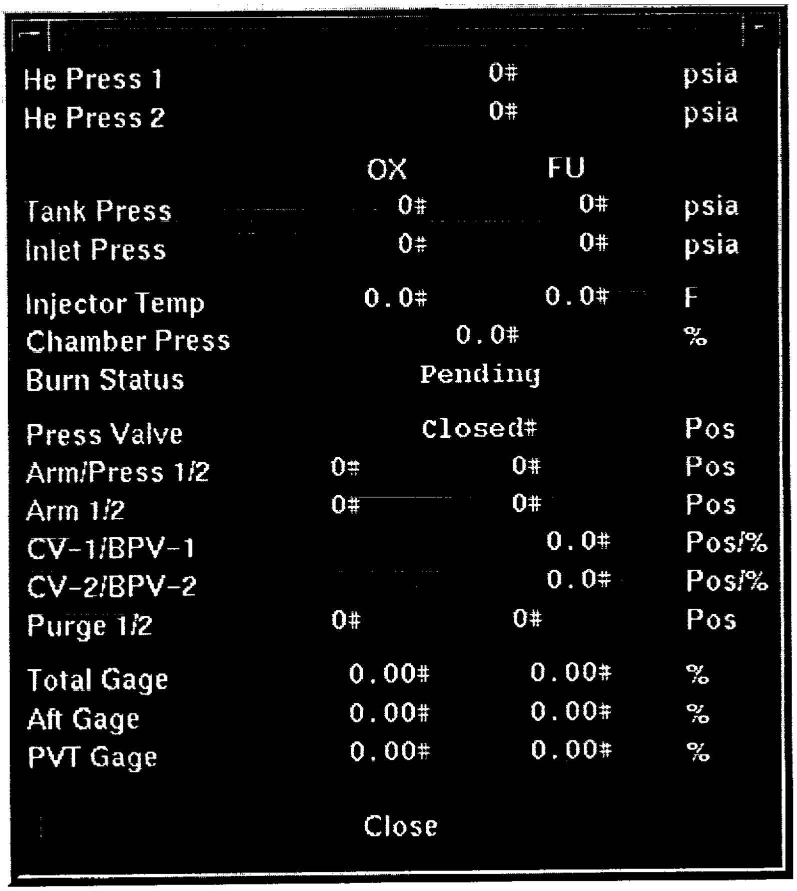
Next, we draw a distinction between two sorts of displays built into this program: *fixed* displays and *adaptive* displays. The fixed displays essentially are conventional flight controller displays showing spacecraft subsystem configurations, current sensor values, internal

[2] Unix is a trademark of AT&T. The X-Windows System is a trademark of Massachusetts Institute of Technology. OSF/Motif is a trademark of Open Software Foundation, Inc. Hugin API is a trademark of Hugin Expert A/S.



	OX	FU	
He Press		0#	psia
Tank Press	0#	0#	psia
Tank Temp	0.0#	0.0#	F
Quantity	0.00#	0.00#	%

Figure 2: Left OMS summary display. Only a small sampling of information about the Left OMS is available to the user from this display.



	OX	FU	
He Press 1	0#		psia
He Press 2	0#		psia
	OX	FU	
Tank Press	0#	0#	psia
Inlet Press	0#	0#	psia
Injector Temp	0.0#	0.0#	F
Chamber Press	0.0#		%
Burn Status	Pending		
Press Valve	Closed#		Pos
Arm/Press 1/2	0#	0#	Pos
Arm 1/2	0#	0#	Pos
CV-1/BPV-1		0.0#	Pos/%
CV-2/BPV-2		0.0#	Pos/%
Purge 1/2	0#	0#	Pos
Total Gage	0.00#	0.00#	%
Aft Gage	0.00#	0.00#	%
PVT Gage	0.00#	0.00#	%

Figure 3: Left OMS detailed display. All of the Left OMS sensor and calculation data is available to the user from this display.

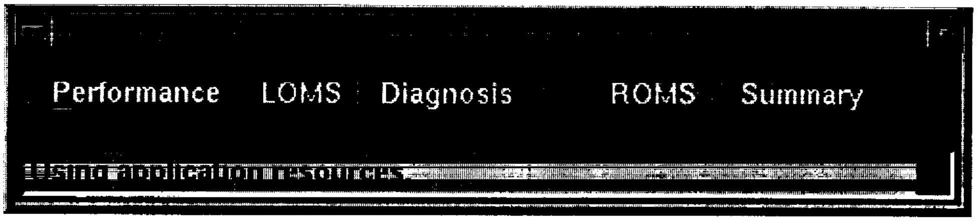


Figure 4: OMS burn program palette menu. The user may select any of the program's displays from this menu, thereby overriding automatic controls over the presented displays. This menu contains pulldown menus for "n-of-many" selections and option menus for "one-of-many" (mutually exclusive) displays.

computation results, mission status information, and so on. These displays are "fixed" because they're compiled into the program. Some of the fixed displays pertain to various levels of "granularity" or detail; these range from showing an overview sampling or summary of important information, to showing every bit of detailed information. In order to manage the "real estate" on the screen, and thereby manage the cognitive load on the user, the program employs the Vista models to select which degree of detail is suitable for display: it chooses the summary displays when there isn't much of interest in the current decision context from one series of displays, but chooses the detailed displays when crucial information from these displays is necessary to make the best-informed decision. If necessary, the program will "shrink" the irrelevant displays and "enlarge" the relevant displays by selecting among the fixed displays in each series. Of course, there may be some information overlap in each level of granularity. Figures 2 and 3 show a summary and detailed display for the Left OMS subsystem. Since we allow the user to override any of these automatic display selections, the program also provides a "palette" menu from which to select any of the displays made available by the program. Figure 4 shows the palette for the OMS burn program.

The adaptive displays provide the users insight into the probability and utility distributions calculated by the inference engine. These displays are "adaptive" in the sense that the program builds them automatically, based on external information, so that various configurations of the displays may be used for different stages of development or by different users. Specifically, the program constructs these displays from information contained within the belief networks; since there are a pair of belief networks for any complete OMS burn model, the program actually builds two sets of displays. First, the program builds a "detailed" diagnosis display which lists all of the possible faults provided by the model. We use a histogram representation to convey the probability distribution over these faults; initially, the distribution corresponds to the *a priori* probabilities of occurrence. Second, the program builds a "summary" diagnosis display which lists all of the fault groups encountered in the fault list. It is assumed that each possible fault is a member of one and only one fault group. Again, we use a histogram representation to convey the probability distribution over the fault groups. As the program acquires and processes telemetry data, the inference engine will determine new probability distributions which the program will present to the user by changing the magnitudes of the appropriate graph elements. Figures 5 and 6 show examples of these displays. These two displays represent the "granularity" offered into

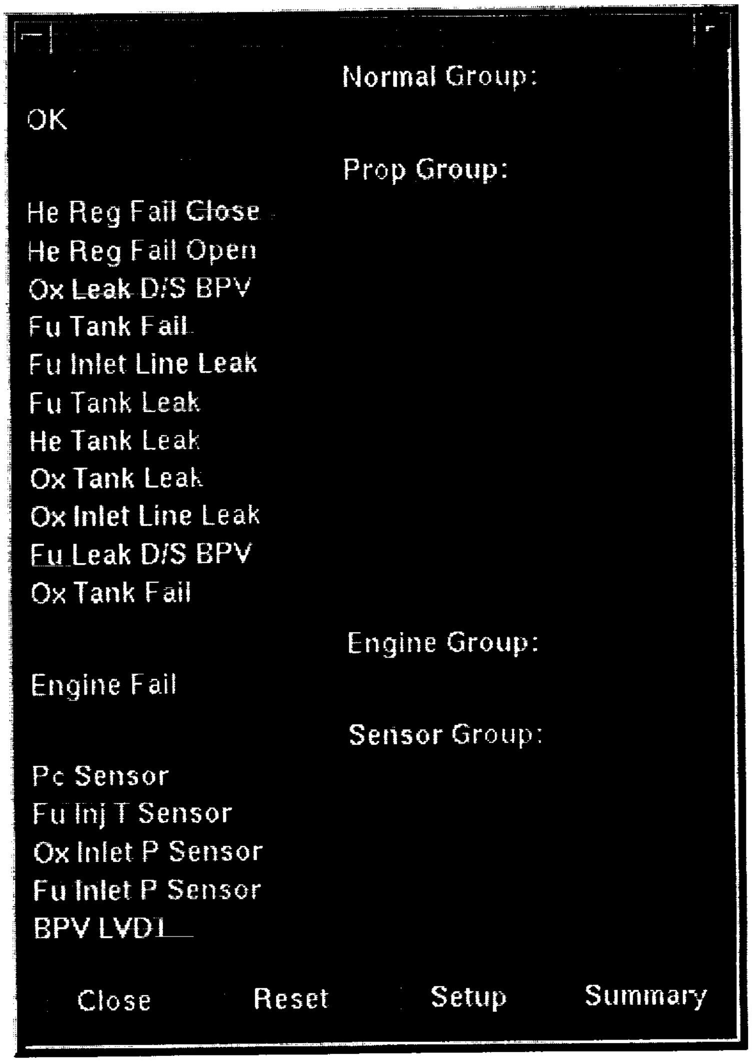


Figure 5: A "detailed" diagnosis display. Each entry in the histogram represents the relative probability for the named fault.

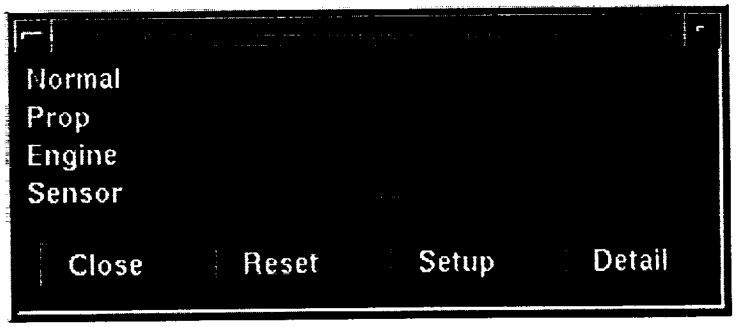


Figure 6: A "summary" diagnosis display. Each entry in the histogram represents the summation of the probabilities for all faults in the named group.

the diagnosis information. Since the summary diagnosis display consumes less screen space than the detailed diagnosis display, it is meant to be used as the primary diagnosis display when the probability of any fault is low. The program will automatically replace the summary display with the detailed display when the probability of any fault exceeds some threshold. We shall describe below a built-in feature which enables the user to adjust this threshold. To override these automatic controls, the displays also provide convenient push-buttons to increase or decrease granularity. There is also a push-button to invoke the "setup" dialog, which we describe below.

Another adaptive display is the *action-selection* display. Since the belief networks do not contain information for the utility models, the program builds this display based on information contained in a user-controlled file. This file contains certain actions and utility model parameters necessary to build the display. Once again, there is one action-selection display for each OMS. Figure 7 shows the action-selection display for the OMS burn monitoring program. Since the number of actions is small in this application, there is only one level of granularity among the action-selection displays.

The "setup" dialog box provides the user with some control over the behavior of the inference and display-management functions (see figure 8). The three option menus provide the user with a mechanism for selecting the context of the OMS burn, such as whether the burn is critical, whether a minimum burn target must be satisfied, and what performance capabilities remain in redundant systems in the event of a failure of the primary system. The configuration of these menus affects the parameters used by the utility model. The "auto-display threshold" slider bar enables the user to select the fault probability value above which the program will automatically present the detailed diagnosis display (for all faults other than "ok"). The "auto-freeze threshold" slider bar enables the user to select the fault probability above which the program will cease to update the probability and utility distributions and displays. This feature disables updates to faults which manifest themselves in a dynamic fashion, presenting evidence convincingly initially (with high probability), then appearing to change into a different signature. Since the initial signature best represents the real problem, we may choose to disable further calculations after exceeding a certain confidence threshold.

Finally, adopting the Vista philosophy on screen real-estate management, the OMS burn program can control the placement of most of these displays automatically. For example, the program will place the Left and Right OMS summary and detailed displays adjacent to each other if a companion display is already visible on the screen. It will also substitute the mutually exclusive displays at the same screen location. These automatic placements override the window manager's controls over window placement. If a companion display is not visible, the program will defer placement to the window manager, which then employs the user's default geometry settings or interactive placement resources. These automatic controls provide convenient display-management techniques which minimize distraction of the user during crucial decision-making contexts.

The OMS burn belief network and utility models capture a tremendous amount of flight controller expertise. The belief networks were developed in direct consultation with flight controllers, and accurately represent the probabilistic reasoning performed by these flight controllers during real-time MCC operations. The *a priori* probabilities for the uncertainty models and the utility parameters for the utility models were derived from

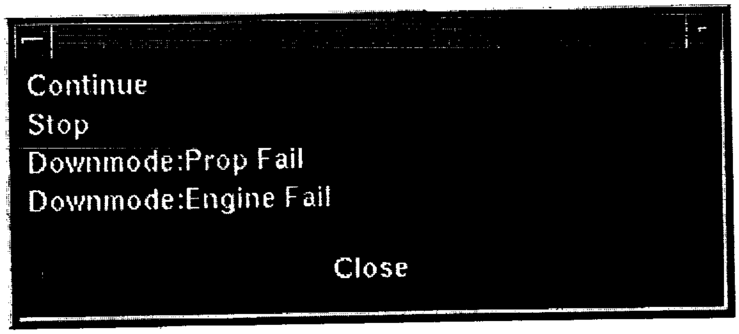
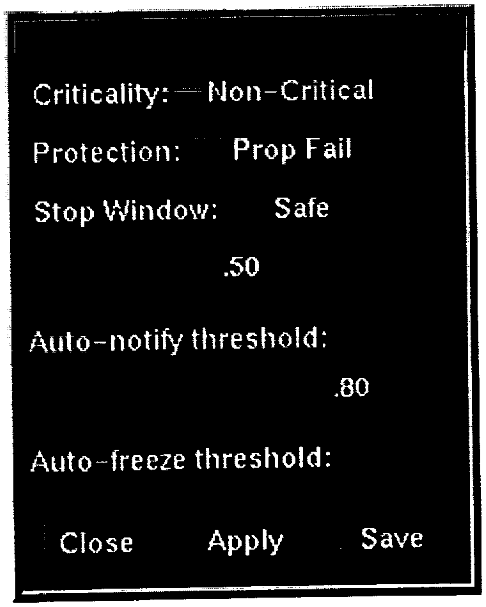


Figure 7: An action-selection display. Each entry in the histogram represents the relative utility for the named action in the current burn context.



Criticality: Non-Critical

Protection: Prop Fail

Stop Window: Safe

.50

Auto-notify threshold:

.80

Auto-freeze threshold:

Figure 8: The OMS burn program "setup" dialog box. Slider bars enable the user to set thresholds for display-management functions. Option menus enable the user to establish the burn context for the utility model.

the results of surveys of all of the flight controllers responsible for OMS burn monitoring. We have found that these model parameters have worked extremely well during rigorous tests of this new program.

4 Future Work

The Vista-I and Vista-II systems have been very successful, particularly in demonstrating the usefulness of these decision-theoretic approaches to decision-making and display-management in real-time operations. These successes have generated many interesting ideas we intend to pursue as we enhance the models and reasoning techniques. Many of these ideas will be pursued during next year's Vista-III project.

Using collaborating Vista models, we are experimenting with a distributed expert system approach to group decision-making applications. Using the *information sharing protocol* developed at JSC [4], we distribute the probability and utility distribution results from various Vista models across a network to other flight controllers whose systems may be affected by the operations of another system. Such a multi-agent application is especially useful for prioritizing a serial list of actions to be forwarded to the astronauts. This approach is also interesting for the deployment of adaptive multi-attribute utility models.

We are also experimenting with the integration of empirical *sensor importance measurements* derived by the *selective monitoring* (SELMON) project at the NASA Jet Propulsion Laboratory (JPL) [5]. These measurements often provide additional intuitive representations of sensor observations as evidence for the sensor nodes in the belief networks, particularly when the dynamic behavior of a sensor is important information.

A focus on sensor importance can also be made from a strictly probabilistic or statistical standpoint. One interesting application of these techniques lies in determining the diminished confidence in the latest sensor reading over time. Another similar application can determine the information content of a particular display, enabling the program to suggest a fixation on that display if it isn't currently visible.

Finally, we are developing new implementation techniques to facilitate the integration of uncertainty models within workstation programs. These implementation techniques include display-management protocols interacting with the window manager, new frameworks of interacting objects to facilitate display construction, and possibly new X-compatible widgets which hide all of these implementation details from the programmer.

References

[1] Horvitz, Ruokangas, Srinivas, and Barry. *A Decision-Theoretic Approach to the Display of Information for Time-Critical Decisions: The Vista Project.* Proceedings of Fourth Annual Workshop on Space Operations Applications and Research (SOAR '92), NASA/Johnson Space Center, Houston, TX, 1992.

[2] Howard and Matheson. "Influence Diagrams," in *The Principles and Applications of Decision Analysis.* Howard and Matheson (eds.), Strategic Decisions Group, Menlo Park, CA, 1981.

[3] Andersen, Olesen, Jensen, and Jensen. *Hugin – A Shell for Building Bayesian Belief Universes for Expert Systems.* Proceedings of the Eleventh International Congress on Artificial Intelligence, 1989.

[4] Barry, Scott and Weismuller. *A Distributed Computing Model for Telemetry*

Data Processing. Submitted to the Goddard Conference on Space Applications of Artificial Intelligence, NASA/Goddard Space Flight Center, Greenbelt, MD, 1994.

[5] Doyle, Chien, Fayyad, and Wyatt. *Focused Real-Time Systems Monitoring Based on Multiple Anomaly Models.* Seventh International Workshop on Qualitative Reasoning, Eastsound, WA, 1993.