Improving Real-time Performance of Intelligent Systems with Dynamic Trade-off Evaluation

U. M. Schwuttke and L. Gasser

Jet Propulsion Laboratory
California Institute of Technology
MS 301-270,4800 Oak Grove Drive,
Pasadena, CA 91109
ums@puente.jpl.nasa.gov

Institute of Safety and Systems Management
University of Southern California
Los Angeles, CA 90089-0021
gasser@usc.edu

ABSTRACT

This paper describes dynamic trade-off evaluation (DTE), a new technique that has been developed to improve the performance of real-time problem solving systems. The DTE technique is most suitable for environments in which the requirement for meeting time constraints is of equal or greater importance to that of providing optimally intelligent solutions. In such environments, the demands of high input data volumes and short response times can rapidly overwhelm traditional AI systems. DTE is based on the recognition that in time-constrained environments, compromises to optimal problem solving (in favor of timeliness) must often be made in the form of trade-offs. DTE combines knowledge-based techniques with decision theory to 1) dynamically modify system behavior and 2) adapt the decision criteria that determine how such modifications are made. The performance of DTE has been evaluated in the context of real-time trade-offs in spacecraft monitoring problems. One such application has demonstrated that DTE can be used to dynamically vary the data that is monitored, making it possible to detect and correctly analyze all anomalous data by examining only an appropriate subset of the total input data. DTE is shown to enhance real-time performance in both conventional and intelligent automation tasks. In carefully structured experimental evaluations that use real spacecraft data and real decision making, DTE provides the ability to handle a three-fold increase in input data (in real-time) with out loss of performance. Such capability makes knowledge-based approaches (which have not typically been been good choices for programming time-constrained applications) more broadly applicable: the knowledge-based techniques in DTE are effective in complex real-time environments and can actually enhance the performance of conventional software.

1.0 Introduction

The greatest challenge to complex systems monitoring is in the area of real-time AI. While speed is relevant to the quest for real-time performance, it is generally agreed that fast systems do not necessarily qualify as real-time systems, The concept of real-time implies timely response under varying input data volumes and under unexpected situations. Real-time systems have therefore been defined as systems in which "there is a strict time limit by which the system must have produced a response to environmental stimuli" [O'Reilly 1985]. According to such definitions, real-time systems must be able to flexibly, reliably, and predictably respond to input data, independent of changes in the rate at which it is arriving. Inherent in this requirement is the need for real-time systems to recognize situations in which optimal problem solving is incompatible with time constraints and respond to these situations by making concessions to optimality with minimum impact on problem solving integrity. One way for intelligent systems to achieve these capabilities is by monitoring aspects of their own performance and, when necessary, evaluating and implementingaeg relevant trade-offs.

As a result, new methods are being sought out for run-time modification of problem solving strategies. These methods aim to provide accurate responses in the presence of time constraints and conflicting objectives [Russell and Wefald 1989], [Horvitz 1988a]. Each of these approaches recognizes the need to make implicit trade-offs and compromises that favor timeliness over optimality, but each has certain limitations for the dynamic real-time environments faced in applications like spacecraft data monitoring. Russell's approach for metareasoning in game playing applications is characterized by the fact that each step of the reasoning process depends on a preceding metareasoning analysis step. However, for many systems it may more appropriate to perform metareasoning only when necessary rather than at every step, particularly when the metareasoning is not time-constrained. Horvitz uses utility analysis to determine the value of a computation prior to deciding whether to initiate that computation or to begin an action. This approach determines the benefit of continuing to refine the solution to a problem, and is a type of incremental (or "anytime") algorithm [Dean 1990]. In severely time-constrained situations, it may be more beneficial to vary the solution strategy according to the dynamics of the environment than to terminate a fixed strategy when dictated by the environment. As a result, these approaches are not ideally suited for complex real-time environments.

Real-time reasoning research has also begun to address specific the issue of trade-offs. For example, research in intelligent data management [Washington and Hayes-Roth 1989] [Hayes-Roth 1990] combines sampling (processing one data point per fixed time interval) and thresholding (processing data points that exceed dynamic threshold levels) to produce a dynamic filtering approach that effectively trades the amount of input data processed for improved timeliness of solution. The specific focus of this technique is on data types that are governed by a set of thresholds or performance specifications.

These approaches are among the first attempts to study simultaneous consideration of time pressure, complex environments, and conflicting objectives. In addition, studies of time-constrained rational agents have begun to emerge in the literature [D'Ambrosio 1990], [Hansson and Mayer 1990]. However, most of these studies rely upon assumptions that are not universally applicable. Game-playing and medical diagnosis, for example, are self-contained domains with (relatively) long time lapses between stimulus and response. Many interesting domains such as spacecraft monitoring are neither self-contained nor free of time constraints. As a result, new methods are needed for complex, highly dynamic applications. Towards this end, a combination of AI and decision theory is proposed.

This paper describes Dynamic Trade-off Evaluation (DTE), a new approach to the real-time reasoning problem that combines decision theory and knowledge-based techniques to automatically determine when trade-offs become necessary and how to implement them with minimal impact on solution quality. DTE offers a general methodology for explicitly making a variety of trade-offs (including the input data management trade-off). This methodology provides the ability to perform metareasoning only when necessary and to dynamically modify the solution strategy based on both the dynamics of the environment and the changing goals of the monitored system.

2.0 Metareasoning with Dynamic Trade-off Evaluation

The applicability of decision theory and the psychology of judgement to the general problem area now known as metareasoning was recognized early, with research on heuristic methods for controlling inference [Simon 1955]. However, initial enthusiasm for using decision theory as an artificial intelligence technique dwindled in favor of other approaches that seemed to lend themselves more naturally to expressing the rich structure of human knowledge [Horvitz et al. 1988b].

Recently, there has been renewed interest in decision theory for real-time AI applications. Rapidly changing circumstances may involve trade-offs and judgements, two processes which can entail a substantial level of subjectivity [von Winterfeldt and Edwards 1986]. Decision theory offers straightforward mechanisms for incorporating subjective evaluations. These mechanisms are embodied in formal decision theoretic principles for obtaining preferred courses of action in the presence of uncertain events and conflicting objectives.

A variety of decision-theoretic techniques exist in multi-attribute utility theory that enable straightforward methods for evaluating competing objectives. However, only three variants of these have been commonly applied to real-world situations [von Winterfeldt and Edwards 1986]: the simple, multi-attribute rating technique [Edwards 1977], difference value measurement [Dyer and Sarin 1979], and subjectively expected utility (SEU) measurement [Keeney and Sicherman 1976]. These techniques are based on general procedures in which a set of possible alternative actions are evaluated with respect to some set of criteria to produce a utility vector. Associated with each action there may be a set of possible outcomes, each of which has some probability of occurrence. The probability and utility vectors can then be combined to produce a value that represents the subjectively expected utility of each action. A decision is made by selecting the action with the maximum utility.

Of the three multi-attribute utility techniques that are commonly applied to real-world problems, the Edwards technique is the simplest computationally, because it uses additive (rather than multiplicative) utility and aggregation models. Furthermore, it relies on direct rating and ratio estimation (rather than probability methods) for determining utilities and weighting factors. As a result, the Edwards technique involves only the calculation of a simple dot-product for each alternative under evaluation. More computationally intensive methods can require the calculation of numerous complex terms in addition to the dot product, for purposes such as evaluating interdependencies among evaluation criteria. However, for many practical applications, the results of the simpler technique are theoretically and behaviorally comparable with the other methods [von Winterfeldt and Edwards 1986].

This section describes a static procedure for decision making in the presence of multiple objectives that is based on the Edwards utility analysis method. It also introduces an approach for automating this method and using it to perform metareasoning in dynamic real-time systems. The simplicity of the Edwards technique offers a significant advantage for real-time application, particularly for problem domains in which it is frequently invoked [Feldman and Sproull 1977].

2.1 Utility Analysis for One-time Multi-attribute Decision Making

Utility analysis methods that are used to make decisions using the principles of multi-attribute utility theory share a common set of procedures for reducing complex evaluation tasks to composites of single attribute decisions. According to these procedures, each single attribute decision is assigned some importance (weight) in the overall decision process. In general, the procedures include the following steps [von Winterfeldt and Edwards 1986]:

1. Definition of application specific alternatives and the criteria that determine the value of these alternatives
2. Separate evaluation of each alternative with regard to each individual attribute (using the specified criteria)
3. Assignment of relative weights to the attributes
4. Aggregation of the single attribute evaluations of alternatives and the attribute weights into an overall evaluation of alternatives
5. Selecting and acting upon the alternative with the maximum value.

The Edwards utility analysis procedure includes straightforward techniques for determining single attribute values, obtaining weighting factors, and forming aggregation models. Single attribute values are derived with direct rating, which is one of the most important and most widely used numerical estimation method for performing value measurement [von Winterfeldt and Edwards 1986]. First, all the alternatives pertaining to a particular situation are specified. Then direct rating is used to rank the alternatives within the specified set: the best and worst possible alternatives are selected from the set and given respective values of 100 and O on a rating scale. The remaining alternatives are ranked at appropriate intervals on the scale so that their positions accurately reflect their perceived value in the subjective judgement of the decision maker(s). The ratio estimation method [Torgenson 1958], [Baird and Noms 1978] is then used to determine the weights that signify the importance of the various alternatives. The weighting scheme provides a representation of the relative importance of the individual attributes to the overall evaluation process. The results of this step again reflect the individual perception of the decision maker(s).

Finally, an additive aggregation model is used to define the overall value of an alternative by summing over the various attributes such that

$$v(x) = \sum_{i=1}^{n} w_i \, v_i(x_i)$$

In this equation, x is the alternative under evaluation, $v_i(x_i)$ is the value of alternative x with respect to the ith attribute, w_i reflects the importance of the ith attribute, and v(x) is the overall aggregate value of alternative x. Using this equation, the various alternatives are evaluated and the one with the greatest aggregate value is selected as the best alternative, according to the most accurate information available at the time the decision is made.

2.2 The Applicability of Utility Analysis to Static Trade-off Evaluation

The utility analysis methods put forth in the general context of multi-attribute utility theory have been studied with respect to one-time decision making and have been applied to such diverse problem areas as selecting real-estate sites [Edwards and Newman 1982] and evaluating coastal development proposals [Gardiner 1974]. Both of these applications are characterized by numerous competing evaluation criteria. Reaching a decision in such applications is a one-time process that relies on utility analysis to evaluate, weight, and aggregate. The decision reflects trade-offs on the pertinent, competing objectives according to the selected weighting factors and incorporates judgement of the decision makers. However, implicit in the procedure is the assumption that the criteria and objectives which formed the basis of the evaluation process will remain valid after a decision has been made, because the decision should not be remade: once selected, a real-estate site or a coastal development proposal should remain appropriate for a suitably long period of time to come, In this type of decision, it is appropriate that the decision criteria, weighting factors, and final evaluation are static in nature.

3.0 Dynamic Trade-off Evaluation for Real-time Problem Solving Systems

In the case of real-time problem solving systems, trade-offs must be made dynamically and continually: a static, one-time decision will not reflect changing circumstances in the application environment. Further, if the application is characterized by changing circumstances, then domain knowledge and judgments may need to be combined with multi-attribute utility theory to enable revision of the decision criteria.

An extension of utility analysis is introduced in this paper as a new approach for real-time decision making that enables dynamic evaluation of real-time trade-offs and maintains the advantages of simplicity, robustness, and flexibility associated with the static method. The new approach is termed Dynamic Trade-off Evaluation (DTE). In DIE, utility analysis is used to rank alternatives in a preference space. Domain knowledge provides decision rules used at run time to 1) dynamically re-weight decision criteria and 2) dynamically select among alternatives in a preference space based on situational attributes and operational modes. DTE is sufficiently general for application to a variety of run-time trade-offs and for integration into a real-time monitoring architecture.

The DTE procedure consists of six steps, some of which are dynamic parallels of steps in the utility analysis procedure. The first three of these steps and part of the fourth must be completed during the design phase of the system; the rest of the steps take place in real-time. The procedure includes:

1. Definition of the trade-off instantiation mechanism. This step involves specif ying the circumstances under which DTE is required and designing the mechanism that will detect those circumstances and invoke the trade-off evaluation.

2. Definition of application-specific alternatives and the attributes or criteria that determine the value of the alternatives. During this step, the alternative actions that are to be considered in the trade-off evaluation are specified, along with criteria that will be used to evaluate the alternatives. As part of this process, the system designers and domain experts also specify domain knowledge and (if necessary) heuristics that define the various ways of implementing each alternative. In addition, the decision

criteria that influence the specific implementation of a run-time alternative are considered,

3. Separate evaluation of each alternative, This is done in conjunction with the previous step, and involves reliance on subjective judgments in cases where no basis for objective evaluation exists. Each alternative is ranked with respect to each of the evaluation criteria, on a scale of O to 100, and suitable consistency checks are applied to the evaluation,

4. Definition of weights and modes. Relative weights are assigned to each of the criteria, along with ranges within which the weights can vary. Domain knowledge is specified to determine the circumstances under which the weights will be varied. In addition, multiple modes may be specified, where each mode is governed by a different set of weights. Both the variation of the weights and the choice of a mode are determined at run time using domain knowledge. These decisions are based on events in the monitored environment.

5. Aggregation. The weights selected in the previous step are used to determine the aggregate value of each of the alternatives, using an additive aggregation model. These aggregate values provide the evaluation of the alternatives with regard to one of the trade-off axes. Depending on the specific trade-off, similar evaluation and aggregation may be required with regard to the second trade-off axis.

6. Selection. An alternative is selected, based on the upper-right-hand-corner criterion specified in static utility analysis methods. When the evaluation indicates that two or more alternatives are equally good, domain knowledge is used to select one alternative over the others, or if the alternatives are not mutually exclusive, to select several of them.

5.0 Applications of Dynamic Trade-off Evaluation

This section describes how dynamic trade-off techniques are applied, using as examples two specific real-world problems in the domain of spacecraft monitoring. The first trade-off that is examined is a timeliness trade-off; specifically, representativeness of the input data vs. timeliness of the solution. This trade-off is examined in the context of spacecraft Solid State Imaging.

A second trade-off that is explored pertains to problem solving strategy. This trade-off weighs focus on a specific problem solving task against general responsiveness to other more important tasks that may arise. The focus vs. responsiveness problem is studied in the domain of Voyager mission system-level analysis. Anomaly analysis at the system level is particularly challenging because it involves coordination of multiple subsystem knowledge base activities. The focus vs. responsiveness trade-off is highly relevant to this problem, particularly because anomalies have been historically categorized with different levels of criticality: focus on resolving low criticality anomalies cannot preclude the detection of subsequent higher level anomalies.

5.1 A Representativeness vs. Timeliness Trade-off

The Solid State Imaging (SS1) Subsystem on newer spacecraft has the capability for much faster image frame times than the technology used on previous missions. Readout rates can be as fast as one image every two seconds (compared to one every 96 seconds previously). This high data volume is particularly noteworthy because imaging data will be returned continuous y over multiple year missions. Associated with each image are numerous engineering parameters that indicate camera status. Examples of this engineering information include dynamic data pertaining to exposure time, filter position, gain state, readout mode, and data compression mode. In addition, there are non-dynamic parameters that indicate general instrument status, voltages and currents.

While non-dynamic parameters could be managed with an extension of the dynamic thresholding technique [Washington and Hayes-Roth 1989], the remaining engineering information does not follow trends and therefore is not well-suited to this approach. This is because the "correctness" of a data value is independent of the correctness of previous values: a value that was correct at one moment can, without changing, become incorrect at the next moment, depending on subsystem goals. For example, exposure times vary with the goals of the subsystem. When goals change, new exposure times maybe required; if the data related to these parameters does not change, system goals will not be achieved. As a result, intelligent management of more complex data requires the application of knowledge-based techniques that incorporate an awareness of the dynamic goals of the monitored system. The large amount of data, the dependence on heuristics, and the complexity of tasks make this an ideal problem for demonstrating dynamic trade-off evaluation.

The basic real-time mission operations task for this subsystem involves comparison of incoming engineering telemetry to a combination of predicted data values or accepted limit ranges, The specific predictions reflect subsystem goals for the planned sequence of system events and the limit ranges reflect the general operating parameters of the instrument. This task involves two AI components in addition to the conventional automation tasks of predicted-to-actual comparison and user-friendly displays and interfaces. The AI components include intelligent data management and anomaly analysis; the latter capability has been addressed previously [Schwuttke et al. 1991] and will not be addressed here.

The (competing) goals of intelligent data management in this application are to dynamically adjust input data volumes to meet processing capabilities while maximizing the resulting information content, maintaining alertness to unusual events in the input data, and focusing on particularly relevant tasks. The DTE procedure is invoked by a software module that analyzes the size of the input backlog, and is applied to this trade-off as described in the remainder of this section. The overall system architecture that provides this capability is shown in Figure 1.

In this architecture, telemetry data is passed through a parameter selector that has the ability to filter desired telemetry channels. If a backlog builds up during either normal monitoring or anomaly detection, the backlog detection module triggers DTE to re-evaluate the channel selection policy. DTE re-evaluation is done on the basis of both preselected alternatives and heuristic reweighting. The reweighing decisions are governed by the current monitoring circumstances and by the mode of operation: monitoring or anomaly detection.

For the solid state imaging application four possible data management alternatives have been specified as a result of extensive interviews with the imaging subsystem specialist. These alternatives are 1.1) eliminating parameters not in the basic monitoring set, 1.2) eliminating parameters not in the minimal set, 2.1) reducing sampling rate on heuristically defined subset of parameters, and 2.2) reducing sampling rate on the entire parameter set. The converse set of alternatives applies when data rates or computational loads from other processes decrease. These converse alternatives include 1.1) adding parameters in the full monitoring set, 2) adding parameters in the basic set, 3) increasing sampling rate on a heuristically defined parameter subset, and 4) increasing sampling rate on the entire parameter set.

The four alternatives are evaluated with regard to criteria that define representativeness. For data

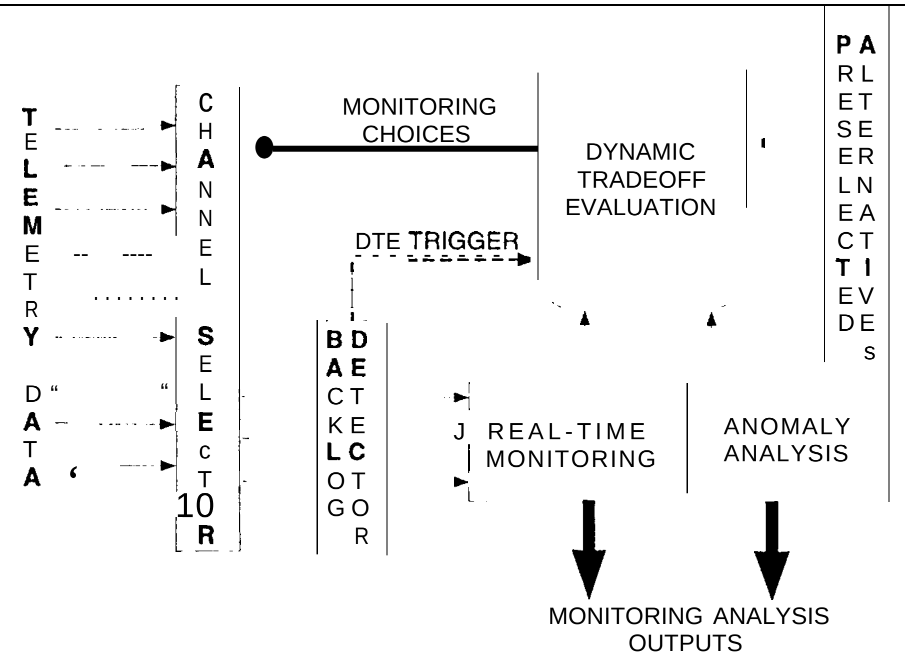

Figure 1. DTE incorporated into a monitoring and diagnosis architecture.

reduction, these include: (A) non-dynamic behavior, (B) irrelevance to an existing problem area, and (C) non-negative impact on monitoring integrity. A data parameter must exhibit non-dynamic behavior before it can be eliminated; frequent parameter value changes indicate a high level of activity that must be monitored to maintain adequate representativeness. When representativeness is an issue, irrelevance to existing problem areas is important in deciding which parameters to remove from the monitored set. Finally, only parameters that do not compromise monitoring integrity in current circumstances can be eliminated without impacting representativeness. Conversely, when the size of the monitoring set is being increased, the criteria become (A) dynamic behavior, (B) relevance to an existing problem area, and (C) positive impact on monitoring integrity.

The second step also requires the specification of domain knowledge that shows how to implement the alternatives. In SS1, the parameter elimination alternatives and the second sampling rate alternative are influenced most heavily by a decision tree that defines which parameter subsets may be deleted from the monitored set. and when they may be deleted. There are also exceptions that apply to deleting some parameter subsets with respect to criterion (A). This exception arises because parameters with significant activity should not be eliminated from the monitored set even if they are part of a predefined deletable subset. In contrast, the heuristically-defined sampling rate alternative is entirely governed by the specific situation in which it is applied. In a normal operating mode, the sampling rate can be reduced on all parameters that are not part of the critical subset. In an anomaly detection mode, the sampling rate can only be reduced on parameters that are irrelevant to anomaly detection. However, in the event of extreme backlogs, reduction on sampling of all parameters may desirable. In these situations it is important to note that if the minimal subset is not preserved, some loss of representativeness may result; domain knowledge must be used to make timeliness vs. representativeness trade-off in these cases.

Occasionally parameters must be added irrespective of the affect on timeliness. This is because in

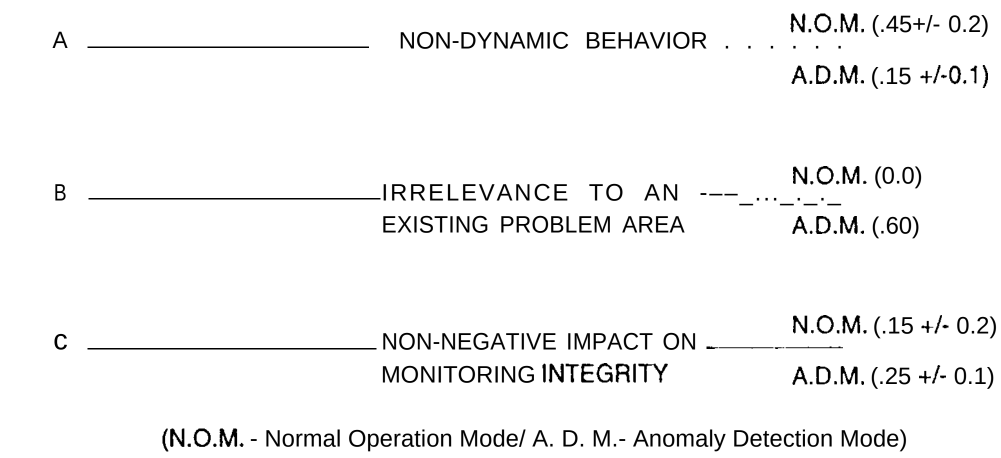

Figure 2. Attributes and weights for input reduction in the SS1 trade-off

anomaly detection mode, increased representativeness might take instant precedence, and parameters pertinent to diagnosing the detected anomaly must be added. With multiple anomalies, more parameters may be needed. Subsequently, timeliness considerations may be applied, and other parameters in the monitored set may be deleted. When the system returns to a normal operating mode, the parameters relevant to a previously resolved anomaly may be removed from the monitoring set if timeliness must be improved.

In the third step, relative weights are assigned to the attributes. Initial weights and variance ranges for these weights are defined so the weights can be adjusted during the reasoning process. This allows the weights to be adapted to accommodate changing circumstances in the monitored environment. Weight variations are initiated when the system detects that its performance is degrading, and are implemented using rules that provide updates based on situational parameters. This step also entails subjectively ranking each alternative in the context of each criterion at design time. The ranking is obtained with the help of the subsystem expert on a scale of O to 100, with 100 having the maximum value, then checked for consistency. For example, alternative 2.1 obviously ranks the highest with regard to B, because the expert specifically designed this alternative not to impact parameters with relevance to an existing problem area. Alternative 1.1, which removes the largest number of anomaly-related parameters, is perceived to be the poorest choice with regard to criterion B. Conversely, when judged against criterion C, alternative 1.1 has the highest ranking because the parameters that it removes generally are the first to be removed and are only added back in small subsets in the event of anomalies.

Two sets of weights are defined for this application, as shown in Figure 2. The first set applies in the normal operating mode and the second applies in an anomaly detection mode. In the normal operating mode, the irrelevance of a parameter to an existing problem area is given no weight, because no problems exist in this mode. However, in anomaly analysis mode, this attribute receives the most weight.

In the fourth step, the single-attribute alternative rankings and the attribute weights are aggregated into an overall evaluation of alternatives which combines with the application-specific domain knowledge to enable the selection of most valuable alternative for the given circumstances. This step differs significantly from the comparable static step for two reasons.

Attribute	Weight*	Weight**	Weight**'	ALTERNATIVE NUMBER 1.1	1.2	2.1	2.2
A	0.15	0.25	0.05	75	90	30	40
B	0.6	0.6	0	20	30	90	50
c	O. 25	0.15	0.35	100	75	40	25
Aggregate Value* (using weight*)				48.25	49.75	68.5	42.25
Aggregate Value'* (using weight**)				45.75	51.75	67.5	43.75
Aggregate Value "** (using wieght***)				50.75	48.75	69.5	33.25

• A.D.M. with no modification on starting weights
•* A.D.M. with weight modification for greater emphasis on environmental dynamics
•** A.D.M. with weight modification for greater empahsis on overall monitoring integrity

Figure 3. Aggregate values of alternatives for varying weights in multiple anomaly mode.

First, circumstances dictate varying weights, which in turn dictate varying aggregations. Secondly, circumstances may vary the knowledge that is applied from situation to situation. Examples of the varying aggregations that are obtained for both operating modes are shown in the tables of Figure 3. These tables show that the data management actions that are most compatible with maintaining maximum representativeness are determined by external circumstances. The ranking of the four alternatives with regard to representativeness value in varying circumstances is summarized in Figure 4, with 1 being the highest ranking.

The final step involves the selection of an alternative based on dynamic evaluation of the representativeness vs. timeliness trade-off. In order to make this trade-off, the four alternatives must also be evaluated with regard to timeliness. The timeliness impact of an alternative is directly proportional to the percentage reduction (or increase) in the number of monitored parameters that results from implementing that alternative. However, this percentage must be calculated immediately prior to making the trade-off based on the parameters in the current set, because the number of monitored parameters is a dynamic quantity determined by the events leading up to

MODE \ ALTERNATIVE	Elimination of than. not in basic subset	Elimination of cham. not in critical subset	Sampling reduction on heuristic subset	Sampling reduction on entire subset
N.O.M. with no modification	1	2	3	4
N.O.M. with backlog modification	2	1	4	3
N.O.M. with monitoring modification	1	2	3	4
A.D.M. with no modification	3	2	1	4
A.D.M. with backlog modification	3	2	1	4
A.D.M. withmonitoring modification	2	3	1	4

Figure 4. Rankings of alternative values with respect to representativeness.

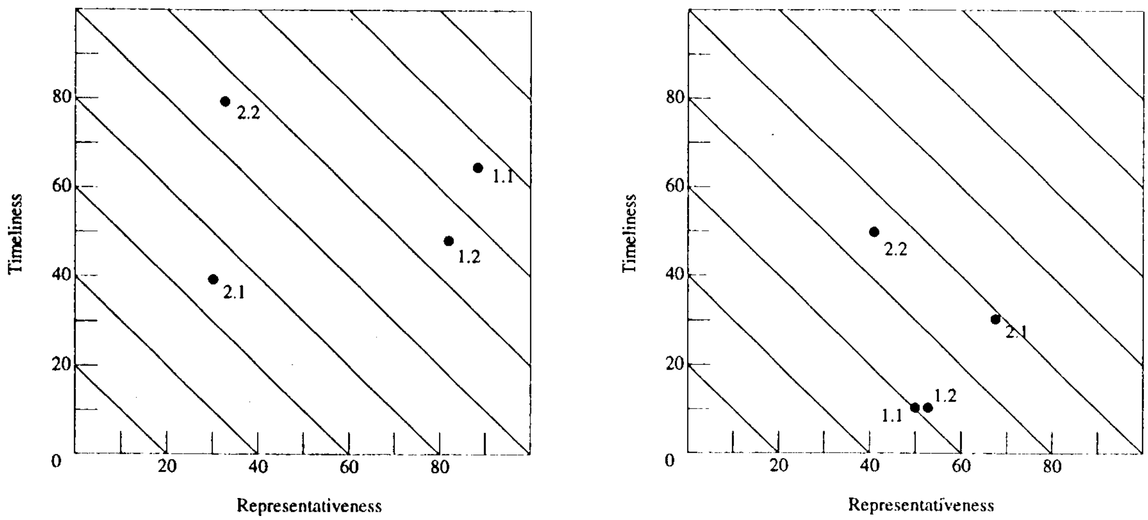

Figure 5. Timeliness vs. Representativeness for Galileo SS1 input data management. Figure (a), at the left, shows the evaluation of the first trade-off in the example discussed in the text, and Figure (b), at the right shows how the dynamics of the environment change after several hypothetical anomalies are detected.

the current circumstances. The following example shows the dynamic and adaptive nature of this evaluation.

Assume that the monitoring system has just been activated. Initially, all 49 parameters are in the monitored set. After some time, the system detects a growing input backlog, and responds by deciding that some parameters must be removed from the monitored set. No anomalies have been detected as yet, and no modifications to the starting weights have been suggested by the knowledge base. As a result, the system uses the aggregate values in the first line of Figure 3 as representativeness values.

Timeliness values are obtained by calculating the net percentage reduction in input data. Alternative 1.1 eliminates parameters not in the minimal set, or 32 of the 49 parameters. Alternative 1.2 eliminates parameters not in the basic set, or 24 of the 49 parameters. These alternatives therefore result in a 65% and a 50% reduction respectively. According to our rules, the reduced sampling alternatives can eliminate 4 out of every 5 input values when no anomalies are present. Thus, with alternative 2.1, we can eliminate 80% of a subset of the monitored set. This subset currently consists of all parameters not in the basic set. A reduction of 80% is therefore possible on 24 of the 49 parameters. With alternative 2.2, we eliminate 80% of the sampling on the entire parameter set, resulting in reductions of 50% and 800/0 respectively. These reductions are plotted against the aggregate representativeness value for each alternative as shown in Figure 5.9 (left). Both representativeness and timeliness are thus rated on a scale of 0-100 and traded off equally; 1 unit on the representativeness scale is equivalent to 1 unit on the timeliness scale. The indifference curves shown in the figure are created by this constant trade-off of units, alternatives lying on the same indifference curve have equivalent value, and alternatives lying nearest to the upper right of the graph are perceived as best. For this application, the alternatives in order of preference are 1.1, 1.2, 2.2 and 2.1. (Note that timeliness considerations have changed the order of preference from that shown in Figure 4, which is based on representativeness alone.)

As a result of this analysis, alternative 1.1 is implemented. Our system is now monitoring only 17 of the 49 parameters, and is achieving adequate throughput. Later, an anomaly appears on parameter 1910, which requires three additional parameters to be added. The anomaly is solved, and at some later time, another anomaly appears on parameter 1881, requiring the addition of 12 more parameters.

We are now actively monitoring 32 parameters, and are building a backlog. The system's backlog detection module initiates metareasoning to reduce it. Figure 5 (right) shows there-evaluation in response to the environmental changes at this point. The analysis proceeded as follows. Alternatives 1.1 and 1.2 will allow only 3 parameters of the 32 parameters being monitored to be eliminated. This is because 12 of the parameters pertain to the current anomaly and 17 belong to the minimal set. Thus, both alternatives achieve a 9.3% reduction in input data. Alternative 2.1 reduces sampling on approximately 60% of the monitored parameters, but because we are in anomaly detection mode, we only filter half of the input data from these parameters, achieving an effective reduction of 30%. With alternative 2.2, we filter half of the input data on all 32 parameters fOr an effective reduction of 50%.

These values are plotted against representativeness as shown in Figure 5. However, the selection of an alternative is not as obvious as previously; alternatives 2.1 and 2.2 are very close to lying on the same indifference curve. This provides an example of the use of domain heuristics in DTE: there is no discernible difference in the value of alternatives 2.1 and 2.2 based on the indiffernece curves alone. However, we can use domain knowledge, which indicates that in the current mode (anomaly analysis), representativeness is a more important considerationthan timeliness (because access to all the information needed to solve the anomaly is more important than saving a little bit of additional time). Thus domain knowledge is used as a tie breaker, and alternative 2.1 with its higher representativeness value, is selected. Eventually, the anomaly on parameter 1881 is resolved, and we return to the normal operation mode. Assuming no change in data rate, in this mode a similar analysis will cause the system to return to its original choice of alternative 1.1, and to continue fully monitoring only parameters in the basic subset. This example has shown the effectiveness of combining utility analysis with domain knowledge to dynamically make real-time trade-offs for intelligent data management. The example illustrates the dynamic nature of the decision environment, and demonstrates the ability to use domain specific heuristics to guide the trade-off process and achieve real-time metareasoning for run-time control.

5.2 A Focus vs. Responsiveness Trade-off

The primary goals of spacecraft mission operations are to design and issue command sequences necessary for enabling mission science goals, to verify correct execution of these sequences, and to monitor health/status. Occasionally, sequence or health/status problems of varying levels of severity are detected. When this occurs, the problems must be analyzed, diagnosed, and resolved as quickly as possible. Until recently, this has been a manually intensive and tedious task.

Recent applications like MARVEL [Schwuttke et al. 1992] and other systems have made contributions towards automated monitoring and diagnosis of complex systems. However, these real-time systems have not addressed the focus vs. responsiveness trade-off pertaining to reasoning modules. In addition, they have not resolved the coordinated analysis of simultaneous anomalies of varying criticality and interdependence. This system-level problem requires coordination of two or more subsystem analysis processes, each of which may have focus/responsiveness decisions.

A IMPACT ON BACKLOG S.A.M. (0.1 +/- .1)
M.A.M. (0.2 +/- .2)

B CONSISTENCY WITH CRITICALITY S.A.M. (0.4)
M.A.M. (0.45)

C CONSISTENCY WITH PROBLEM SOLVING STATUS S.A.M. (0.5 +/- .1)
M.A.M. (0.35 +/- .2)

(S. A.M.- Single Anomaly Mode/ M.A.M. - Multiple Anomaly Mode)

Figure 6. Attributes and weights for Voyager anomaly analysis trade-off

The problem domain for the focus vs. responsiveness trade-off is an environment in which telemetry is monitored (compared to expectation for the purpose of detecting anomalies) by a procedural software module. Anomalous data that is detected by the monitoring module is passed to a knowledge-based system for analysis. The analysis process is significantly slower than the monitoring process; as a result, when several anomalies occur within a very short period of time, an unpredictable backlog can form at the analysis process. Dynamic trade-off evaluation can be invoked to determine how to optimize reasoning in the presence of an anomaly backlog. The relevant alternatives that must be evaluated for the focus vs. responsiveness trade-off include 1) manipulating duration or frequency of input scanning, 2) continuing to analyze the anomaly under investigation, 3) saving state and investigating a higher priority anomaly, and 4) abandoning state and investigating a higher priority anomaly. These alternatives will be evaluated with respect to A) impact on timeliness, B) consistency with criticality, and C) consistency with problem solving status. There are three possible problem solving status cases. These are C1) that the reasoning module is idle while waiting for additional input, C2) that the reasoning module has almost completed its current reasoning task, or C3) that it has not yet neared completion of its current reasoning task. In addition, there are three modes. These are the normal operations mode, the single anomaly mode, and the multiple anomaly mode. However, the normal operation mode reduces to a trivial alternative case, because there are no anomalies to analyze and the system can focus exclusively on anomaly detection. As a result of this simplification, only the latter two

		ALTERNATIVE NUMBER			
ATTRIBUTE		1	2	3	4
A		70	40	30	10
B		40	50	60	70
C	C1	40	80	70	10
	C2	60	10	70	55
	C3	20	30	40	60

Figure 7. Values of the lternatives fo the Voyager monitoring trade-off.

				ATTRIBUTE C1				ATTRIBUTE C2				ATTRIBUTE C3			
				ALTERNATIVE				ALTERNATIVE				ALTERNATIVE			
				1	2	3	4	1	2	3	4	1	2	3	4
Attribute	Weight*	Weight**	Weight"*												
A	0.2	0	0.4	70	40	30	10	70	40	30	10	70	40	30	10
B	0.45	0.45	0.45	40	50	60	70	40	50	60	70	40	50	60	70
c	0.35	0.55	0.15	60	10	70	55	40	80	70	10	20	30	40	60
Aggregate Value* (using weight*)				53	34	57.5	52.75	46	58.5	57.5	37	39	41	47	54.5
Aggregate Value** (using weight**)				51	28	65.5	61.75	40	66.5	65.5	37	29	39	27	64.5
Aggregate Valu-*** (using weight***)				55	40	49.5	43.75	52	50.5	49.5	37	49	43	45	44.5

"M. A.M.. no modification on starting weights "'M.A.M. emphasis on problem status ***M.A.M. emphasis on timeliness

Figure 8. Aggregate values of alternatives in single (top) or multiple (bottom) anomaly mode.

modes need to be considered further. The modes and weights pertaining to these attributes are shown in Figure 6. The table in Figure 7 shows the evaluation of the four alternatives with respect to each attribute. Only one of the rows (C1-C3) pertaining to problem solving status is selected when the trade-off evaluation is performed. This selection is made based on the actual status at the time of the trade-off.

The aggregate rankings of the various alternative are obtained as they were in the example of the previous section. Figure 8 shows the rankings for each of the anomaly modes, problem solving statuses, and weights. Figure 9 summarizes the variation in alternative rankings based on responsiveness factors alone.

As in the previous example, the value associated with a particular alternative based on only one

PROBLEM SOLVING STATUS	MODE	ALTERNATIVE			
		1	2	3	4
C1	S.A.M. with no modification on starting weights	3	4	1	2
	S.A.M. with emphasis on problem solving status	3	4	1	2
	S.A.M. with emphasis on timeliness	2	4	1	3
	M. A. M..with no modification on starting weights	2	4	1	3
	M.A.M. with emphasis on problem solving status	3	4	1	2
	M.A.M. with emphasis on timeliness	1	4	2	3
C2	S.A.M. with no modification on starling weights	3	2	1	4
	S.A.M. with emphasis on problem solving status	3	1	2	4
	S.A.M. with emphasis on timeliness	3	1	2	4
	M.A.M..with no modification on starting weights	3	1	2	4
	M.A.M. with emphasis on problem solving status	3	1	2	4
	M.A.M. with emphasis on timeliness	1	2	3	4
C3	S.A.M. with no modification on starting weights	4	3	2	1
	S.A.M. with emphasis on problem solving status	3	2	4	1
	S.A.M. with emphasis on timeliness	4	3	2	1
	M. A. M..with no modification on starting weights	4	3	2	1
	M.A.M. with emphasis on problem solving status	3	2	4	1
	M.A.M. with emphasis on timeliness	1	4	2	3

Figure 9. Rankings of responsiveness alternatives with respect to mode.

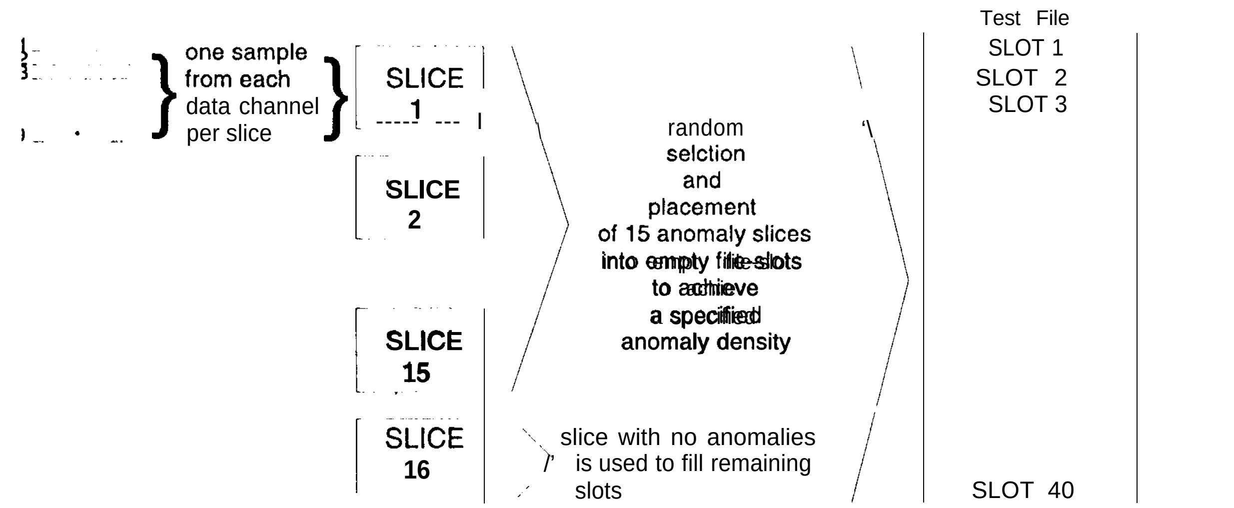

Figure 1O. Test filecomposition.

trade-off dimension (in this case representativeness) varies considerably with the circumstances in the monitored environment. The table of Figure 9 shows that each of the 4 alternatives maybe ranked in any position (1 through 4), depending on the dynamics of the situation.

6.0 IMPLEMENTATION AND TESTING OF DYNAMIC TRADE-OFF EVALUATION

The DTE methods outlined in the previous section have been successfully applied to the dynamic evaluation of the SS1 timeliness vs. representativeness trade-off. This section describes the procedures used to evaluate the DTE mechanism, presents the results that have been obtained, and provides a discussion of the results.

DTE has been designed and implemented as described in Section 5.2. The actual DTE algorithms have been implemented separately from the domain knowledge required to perform application-specific evaluation of trade-offs, so that the DTE modules can be reused in future applications. This will only require the specification and addition of alternatives, evaluation attributes, weights, and rules that are needed by the specific applications.

The purpose of this implementation has been to explore the validity of the approach. Data reduction of approximately 67% was expected to be achievable with the DTE method in the SS1 application domain, because the minimal monitoring set as defined in the domain knowledge is approximately one third of the total parameter set. (When the number of parameters drops below the minimal monitoring set, some loss of monitoring integrity is expected.)

6.1. Experimental Design

Data from Galileo's first earth encounter was archived for use in developing the monitoring, analysis and DTE modules that will be required for an operational system. Initially, we had planned to use this data in its original form for testing DTE's performance. However, the anomaly density from this short encounter was too low to provide adequate representation of the broad range of operational conditions that are expected in the future. Therefore, the DTE module has been tested with randomly generated anomaly files that are composed as shown in Figure 10.

The test files were generated from a base of 16 sample data slices. Each slice contains one sample from each of the 49 SS1 telemetry lines, paralleling the arrival of data in an operational system. A single telemetry line sample is composed of a parameter number, a time tag, a parameter mnemonic, and a data value. An anomaly consists of one or more telemetry lines in which the data values differ from predictions. Fifteen of the sample slices contain actual anomalies that have occurred in the past; these have been supplied by JPL's imaging-subsystem expert. The sixteenth slice contains no anomalous data.

The 16 data slices were used as building blocks by a software program that was implemented to automatically generate randomized test files. Each test file is composed of 8,000 lines of telemetry data, or 160 slices. Most of the slices in a test file are replications of the single anomaly-free telemetry slice. The test files are randomly seeded with selections from the **15** anomalous data slices, in varying anomaly densities. The selected densities include 0%, 30/., 50/., 10%, 15%, 20%, and 30°/0. These densities were chosen to provide evaluation of the DTE methods across the complete range of possible anomaly densities in foreseeable missions, in order to enable insight into applications and situations for which DTE is most successful. For each of the selected anomaly densities, 3 different test files were generated. The entire set of test files was then used to supply data to DTE and two other data management approaches. The other approaches include random data elimination and incremental filtering. These methods provide a means for comparing DTE to previously available approaches. Random data elimination may appear to be an unusual choice. However, it is the method that most closely parallels the approach actually used by many human analysts in data-overload situations: when their data backlog becomes too large, they skip over the data in the backlog, and focus on the newly arriving data. Incremental filtering, on the other hand, involves less loss of information. Data is filtered at according to $f = n/b$ (when $n<b$), where f is the fraction of samples passed through the filter, n is the total number of parameter types, and b is the number of samples in the backlog.

The backlog accrues according to the ratio of the incoming data rate to real-time processing rate. (For example, with a backlog accrual of x, data is arriving x times faster than it can be processed.) Each of the three methods were evaluated with respect to performance under increasing backlog accrual. Two performance criteria are identified: percentage of anomalies successfully detected and percentage of data processed that is needed for correctly diagnosing anomalies. The latter data is referred to as anomaly-relevant data and includes both the anomalies and the information related to solving the anomaly. Successful detection of an anomaly involves perceiving the anomaly within four slices of its occurrence.

6.2. Experimental Results

The experimental results were evaluated with respect to average percentage of anomalies successfully detected under increasing input data volumes and increasing anomaly density and also with respect to anomaly-relevant information that is processed under increasing input data volumes and increasing anomaly density. The first of these evaluation criteria is important with respect to automated monitoring, and the second is relevant to automated anomaly analysis.

Anomaly detection using DTE is highly successful for data rates as high as 2.5 times the real-time monitoring capability, particularly at anomaly densities of 10% or less. In these operating ranges, DTE outperforms both random data elimination and incremental filtering, detecting over 90% of all anomalies within 4 data slices of their occurrence. The success of anomaly detection with random data elimination and incremental filtering, on the other hand, drops below 50% at backlog accrual rates as low as 1.5.

Processing of anomaly-relevant information involves passing parameters relevant to the analysis of a detected anomaly from the data management module to the monitoring and analysis module. If anomaly-relevant parameters are being filtered by the data management module, some of the information needed for analysis will be lost, Intelligent data management with DTE is most successful at low (5% or lower) anomaly densities with backlog accrual rates that exceed real-time processing capabilities by as much as 2.5. Within these operational parameters, processing of anomaly-relevant information is as high as 95% for backlog accrual rates equal to twice the processing capability, 80% for backlog accrual rates equal to 2.5 times the processing capability, and 70% for backlog accrual rates equal to three times the processing capability. At these backlog accrual rates, the other two methods provide no more than 70%, 50%, and 509'. of anomaly relevant data, respectively.

6.3. Discussion

Three important characteristics of intelligent data management systems have been identified [Washington and Hayes-Roth 1989].

- The system should be responsive to changing resource requirements. For example, the amount of data sampled should vary with the computational load placed on the system.
- The system should be responsive to important and unusual events in the input data, even when it is “busy”.
- The system should be able to focus its attention on parameters that are particularly relevant to the current reasoning task,

The Galileo SS1 application has shown that DTE provides an effective way to achieve each of these criteria, not only for the data addressed by Washington that is monitored based on known limits, but also for dynamic data types that do not occur in his application.

DTE enables both anomaly detection and anomaly diagnosis for low anomaly densities and moderate backlog accrual rates. Actual anomaly densities for this application average less than 3%, which is well within the acceptable operational parameters of the method.

Both Figures 11 and 12 show performance degradation in the DTE method beginning at backlog accrual rates that exceed real-time processing capability by a factor greater than three. Furthermore, when the backlog accrual rates exceed processing rates by a factor of four or more, the DTE method begins to converge with the other two methods. This performance degradation is inherent in the domain. The minimal monitoring set for complete anomaly detection (as defined by the domain expert) consists of 17 parameters, or one third of the entire parameter set. When all of these 17 parameters are not monitored, some loss of monitoring integrity will occur, as is demonstrated by degradation of the DTE method at backlog accrual rates of 3.5 or more.

Subsequent testing shows that with an imaginary domain, in which the minimal set can be defined as a significantly smaller subset of the total parameter subset, the effective increase in data reduction that can be achieved is on the same order as the decrease in size of the minimal set.

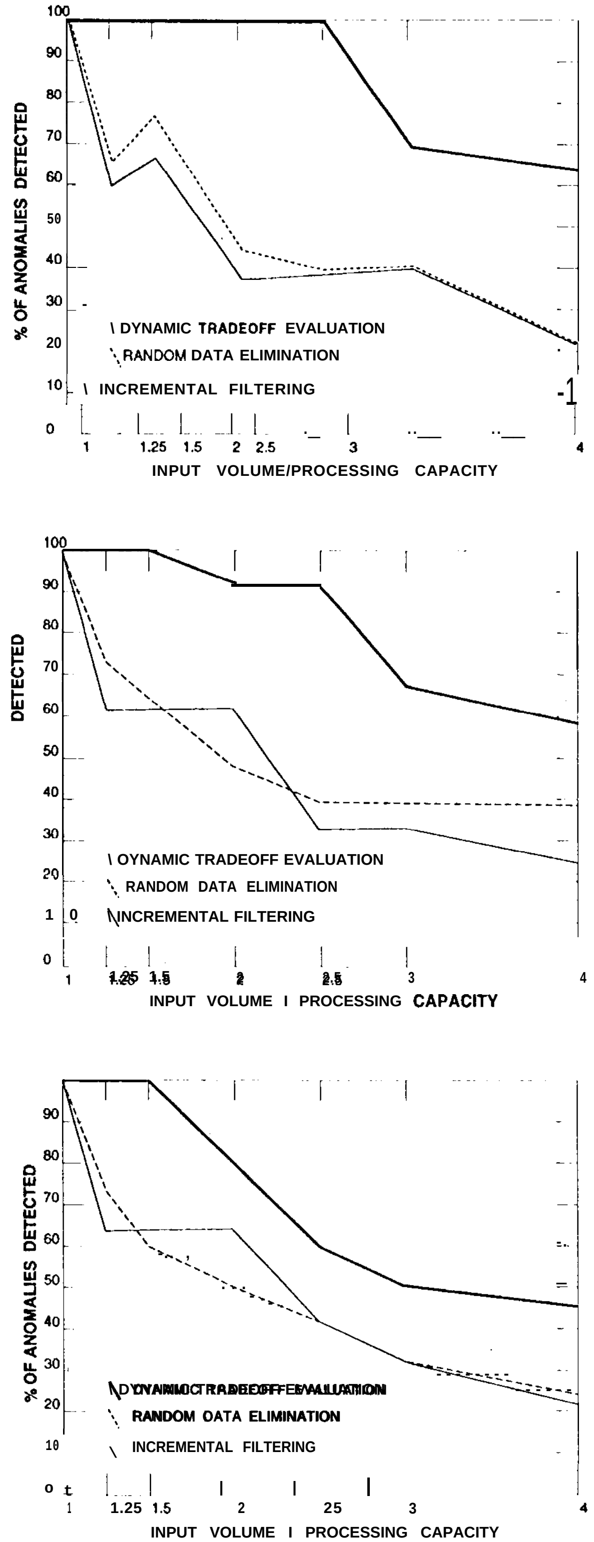

Figure 11. Anomaly detection results of DTE applied to input data management for 3% anomaly density (top), 10% anomaly density (middle), and 20% anomaly desity (bottom).

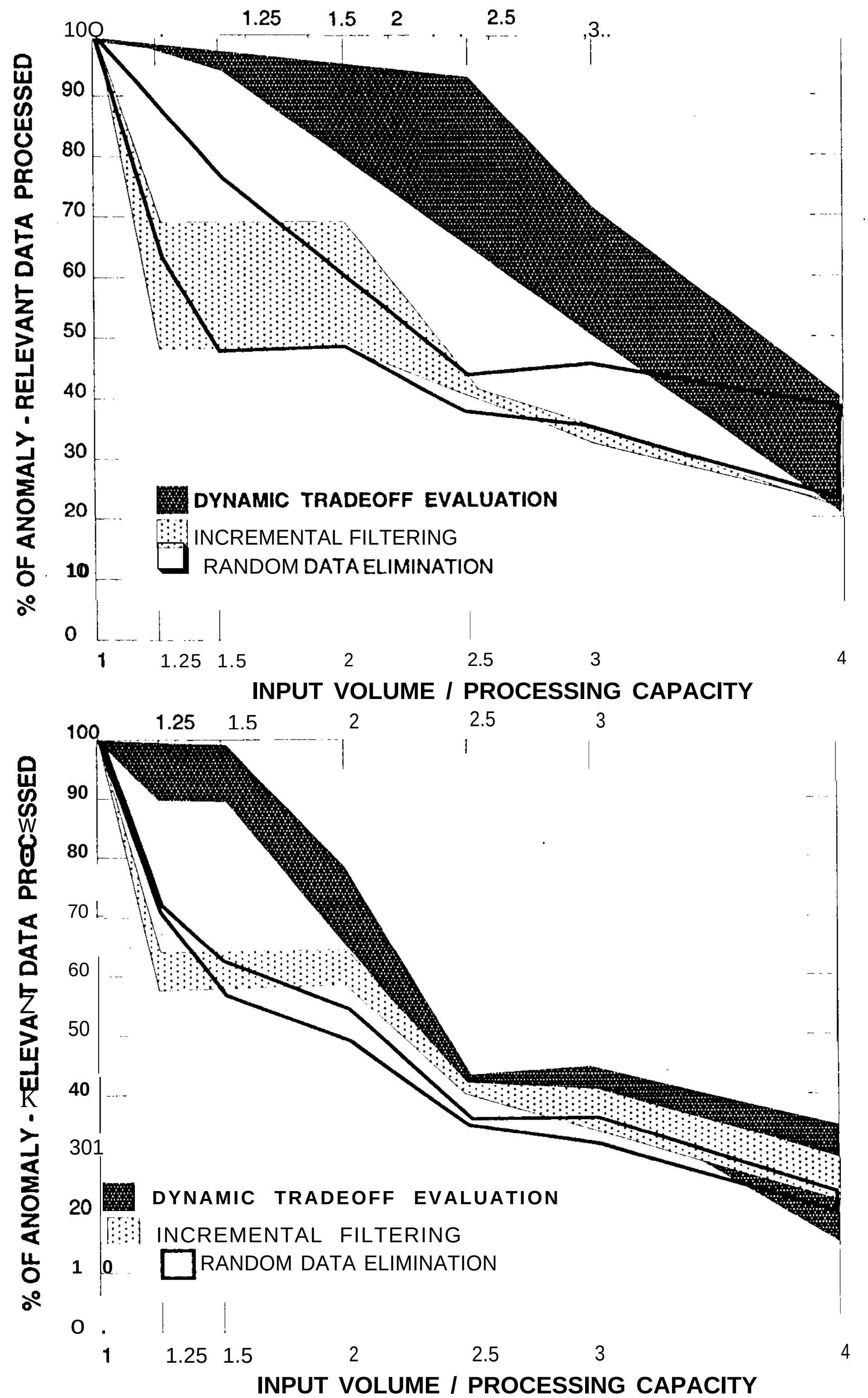

Figure 12 Anomaly processing results applied to input data management for 5% anomaly density (top), and 20% anomaly density (bottom).

A long-term solution to this problem in the context of a specific domain involves designing telemetry (or other input data) definition to later enable maximum data reduction. The more hierarchically the monitored data can be structured, the more the size of the minimal data set can be reduced. However, this type of telemetry design requires monitoring needs to be explicitly considered during spacecraft (or other system) design. As monitoring applications continue to increase in complexity, such an approach would involve significant monitoring benefits in the form of reduced automation and monitoring workforce costs.

The observed performance degradations at known data reduction ratios can be used to enable the system to predict its own failure and provide warnings when its monitoring integrity will be reduced. For example, when the system detects that it has reduced its monitoring coverage below the minimal data subset it can, as a minimum, notify human analysts of possible reductions in monitoring integrity. In a distributed environment, a module that can predict its own failure to meet real-time constraints could actually go so far as to request additional processing resources from the environment.

The significance of the observed improvements in anomaly detection and intelligent data management varies with the significance of the anomaly. An actual SS1 anomaly that occurred within the past year provides effective illustration of this point. In this anomaly, the SS1 camera continued shattering (taking pictures) for 16 hours longer than expected. There was no automated monitoring capability to detect this problem, so it went undetected for several hours. When finally discovered by an analyst, it was not detected at its origin but in one of its symptoms. As a result, the diagnosis of the problem first headed down a false path, adding additional hours of analysis beyond what would have been required if the problem had been detected at its origin. According to subsystem experts, an automated monitoring capability would have saved 3 hours of discovery and 5 hours of analysis, exactly one half of the sixteen hour process that was required in the absence of automated monitoring. An automated analysis capability, or even an intelligent data management scheme that presented relevant. data to the analyst in an organized fashion, could have saved even more analysis time.

The significance of this savings is considerable. The total lifetime of the camera is 150,000 shutters, of which 40,000 were expended prior to launch. Of the remaining 110,000 shutters, 1000 were wasted as a result of this anomaly. With an automated data management and monitoring capability, at least 500 of these shutters, or almost .5% of the life of the camera would have been saved. The value of the possible science return of the additional photographs cannot easily be quantified. However, this example effectively illustrates the fact that any improvement in automated monitoring and analysis can prove to be highly significant in the face of unanticipated future anomalies.

CONCLUSIONS

Dynamic Trade-off Evaluation has been shown to be an effective technique that offers significant benefit to real-time AI systems. DTE incorporates a mix of knowledge-based and utility-theoretic techniques and is particularlyvaluable in real-time monitoring situations of moderate anomaly densities, varying data rates, and dynamic decision criteria. In experimental evaluations, DTE significantly outperforms other commonly-used approaches to manage real-time monitoring data trade-offs in increasing-backlog situations. Moreover, DTE is a generic technique that can be effectively applied in many kinds of trade-off analysis for real-time systems We have designed a generic architecture for DTE applications, treated elsewhere [Schwuttke

1991a], and have taken initial steps to implement DTE as an operational part of the MARVEL intelligent monitoring system [Schwuttke et al. 1991; Schwuttke et al. 1992] in use at JPL.

ACKNOWLEDGEMENT

The research described in this paper was carried out by the Jet Propulsion Laboratory, California Institute of Technology under a contract with the National Aeronautics and Space Administration.

REFERENCES

Baird, J. C. and Noms, E. 1978. "Fundamentals of Scaling and Psychophysics." New York; Wiley Publishers.

D'Ambrosio, B. 1990. "Constrained Rational Agents." *IEEE Transactions on Systems, Man, and Cybernetics.*

Dean, T. 1990. "Decision-Theoretical Control of Inference for Time-Critical Applications." *Technical Report CS-90-44,* Brown University Dept. of Computer Science, Providence, Rhode Island.

Dyer, S. and Sarin, R. A. 1979. "Measurable Multi-attribute Value Functions." *Operations Research, 22, 810-822.*

Edwards, W. 1977. "HOW to Use Multi-attribute Utility Measurement for Social Decision Making." *IEEE Transactions on Systems, Man and Cybernetics*, SMC-7, *326-40.*

Edwards, W. and Newman, J. R. 1982. "Multi-attribute Evaluation." Beverly Hills, CA: Sage Publishers.

Feldman, J. A. and Sproull, R. F. 1977. "Decision Theory and Artificial Intelligence II: The Hungry Monkey." *Cognitive Science 1,158-192.*

Gardiner, P. 1974. "The Application of Decision Technology and Monte Carlo Simulation to Multiple Objective Public Policy Decision Making: A Case study in California Coastal Zone Management." Unpublished Doctoral Dissertation, University of Southern California, Los Angeles.

Hansson, O. and Mayer, A. 1990. "Probabilistic Heuristic Estimates." *Annals of Mathematics and Artificial Intelligence,* 2:209-220.

Hayes-Roth, Barbara. 1990. "Architectural Foundations for Real-time Performance." *Real-time* Systems, May 1990.

Horvitz, E. J. 1988a, "Reasoning Under Varying and Uncertain Resource Constraints." *Proceedings of the National Conference on Artificial Intelligence,* Minneapolis, MN August 1988.

Horvitz E. J.; Breese, J. S, and Henrion, M. 1988b. "Decision Theory in Expert Systems and Artificial Intelligence." *International Journal Of Approximate Reasoning, Special Issue on Uncertain Reasoning: 247-302.*

Keeney, R. L. and Sicherman, A. 1976, "An Interactive Computer Program for Assessing and Using Multi-attribute Utility Functions." *Behavioral Science,* 21, 173-182.

O'Reilly, C.A. and Cromarty, A.S. 1985. " 'Fast' is not 'Real-time': Designing Effective Real-time AI Systems." Applications of Artificial Intelligence II, *SPIE 548,249-257.*

Russell, S. J. and Wefald, E. H. 1989. "On Optimal Game-Tree Search Using Rational Metareasoning," International Joint Conference on Artificial Intelligence, Detroit, Michigan, 334-340.

Schwuttke, U. M.; Quan, A. and Gasser, L. 1991. "Improved Real-time Performance in Automated Monitoring Using Multiple Cooperating Expert Systems" *Proceedings of 4th International Conference on Industrial and Engineering Applications of Al and Expert Systems,* June 1991.

Schwuttke, U. M. 1991a. "Intelligent Real-time Monitoring of Complex Systems." Ph.D. dissertation, Dept. of Electrical Engineering, University of Southern California.

Schwuttke, U. M.; Quan, A. G.; Angelino R.; et al. 1992. "MARVEL: A Distributed Real-time Monitoring and Analysis Application," In *Innovative Applications of Artificial Intelligence 4,* eds. P. Klahr and C. Scott. Boston: MIT Press.

Simon, H. A. 1955. "A Behavioral Model of Rational Choice." *Quart.]. Econ., 69,99-118.*

Torgenson, W. S. 1958. "Theory and Methods of Scaling." New York: Wiley Publishers.

von Winterfeld, D. and Edwards, W. 1986. "Decision Analysis and Behavioral Research." Cambridge University Press.

Washington, R. and Hayes-Roth, B. 1989. "Input Data Management in Real-time AI Systems." Eleventh International Joint Conference in Artificial Intelligence. Detroit, Michigan.