N93-32139

Attention Focussing and Anomaly Detection in Real-time Systems Monitoring

Richard J. Doyle, Steve A. Chien, Usama M. Fayyad, and Harry J. Porta
Artificial Intelligence Group
Jet Propulsion Laboratory
California Institute of Technology
Pasadena, CA 91109-8099

Abstract

In real-time monitoring situations, more information is not necessarily better. When faced with complex emergency situations, operators can experience information overload and a compromising of their ability to react quickly and correctly. We describe an approach to focusing operator attention in real-time systems monitoring based on a set of empirical and model-based measures for determining the relative importance of sensor data.

Introduction: Sensor Selection

Mission Operations personnel within NASA are beginning to face the manifestations of a technology race. Our ability to devise safe, reliable monitoring strategies is not keeping pace with our ability to build space platforms of increasingly complex behavior with large numbers of sensors. To date, spacecraft such as Voyager have had sensor complements numbering only in the hundreds. For these space platforms, it has proven both feasible and appropriate to adopt a comprehensive monitoring strategy where mission operators interpret all of the sensor data all of the time.

However, NASA is moving into an era where sensors on space platforms such as Space Station Freedom will be numbered in the thousands. With space platforms of this complexity, the comprehensive monitoring strategy will be no longer tenable. This trend is not unique to NASA.

It is our thesis that for complex systems with large sensor complements a selective monitoring strategy must be substituted for the comprehensive strategy. The subject of our work is an approach to determining from moment to moment which subset of the available sensor data for a system is most informative about the state of the system and about interactions occurring within the system. We term this process *sensor selection* and we have implemented a prototype selective monitoring system called SELMON [Doyle and Fayyad 91, Chien et al 92, Doyle et al 92].

The SELMON system has its origins in a sensor planning system called GRIPE [Doyle et al 86] which planned information gathering activities to verify the execution of robot task plans. The goal of the current SELMON project is to provide assistance to operators by focusing their attention during real-time monitoring. Our sensor selection approach also could be embedded as part of an autonomous monitoring and control system.

Approach: Sensor Ordering

Our approach to focusing operator attention in real-time monitoring involves defining a set of sensor scoring measures. Each of these measures embodies a different viewpoint on why, at a particular moment, one sensor may be more worthy of operator attention than others. The measures are based in concepts from model-based reasoning and information theory. Some of the measures utilize sensor value predictions generated by simulating a causal model of the system being monitored.

During each timestep all sensors are scored according to these measures. The scores are used as a basis for an ordering on the sensors. See Figure 1. These scoring measures are divided into two categories. The first set – empirical methods – rely upon current and historical data to determine importance. These measures include *surprise*, *alarm*, *anticipate alarm*, and *value change*. The second set uses a causal model of the system to reason about expected current and future performance of the system to determine sensor importance. These methods include *deviation*, *sensitivity*, and *cascading alarms*.

After describing each of these measures, we describe how these measures are combined into an overall importance score for each sensor.

Empirical Sensor Scoring

In this section, we describe the empirical measures that are used in determining the overall importance score assigned to each sensor. This part of the score is based on four measures: *surprise*, *alarm*, *anticipate alarm*, and *value change*. These measures use knowledge about each individual sensor, independently of any knowledge about the interconnectedness of the sensors.

Surprise In order to obtain an ordering on the set of sensors, we need to quantify the following notions: How reliable is a sensor? How stable is it? How often does it go into an alarm state?

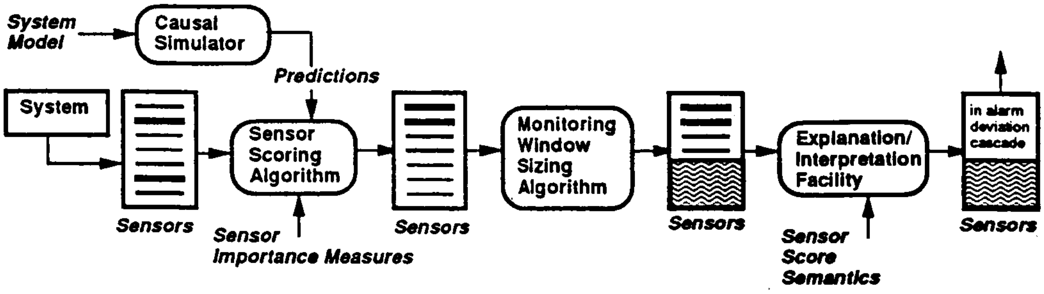

Figure 1: SELMON Architecture.

From an information theoretic point of view, a change in the value of a sensor gives us a certain amount of information (usually measured in bits). Assume we have two sensors, S_A and S_B. Further assume that sensor S_A's value has been wildly changing over the last 100 readings, while sensor S_B's value has been constant. If we are told that according to the latest update, the values of both sensors have increased by 25%, which do we consider a more informative event? Clearly the fact that S_B's value changed is more informative since it is more unusual. Prior to the latest reading, if we were asked to predict the values of S_A and S_B, then based on previous data, we would naturally guess that S_A's value is likely to have changed while S_B's value is likely to have remained constant. Then the fact that S_B changed value tells us something that we did not know or expect.

For each sensor, a cumulative histogram of its values is maintained for each system operating mode. This is done by dividing its range into a fixed number of bins. The boundaries between bins are determined through specific knowledge of the sensor and of the "interesting" subranges in its range. This histogram is then used to determine two measures of the interestingness of the most recent value returned by a sensor.

Denote the range of sensor S by $Range(S)$. If S is a continuously valued sensor, we can discretize its range into a set of collectively exhaustive ranges $\{R_1(S), R_2(S), \ldots, R_K(S)\}$, where

$$Range(S) = \bigcup_{i=1}^{K} R_i(S)$$

With each range $R_i(S)$ we associate a frequency measure $f_i(S)$ that gives the proportion of time that S's value has been in this range. Thus $f_i(S)$ is an estimate of the probability of the value of S falling in range $R_i(S)$ and

$$\sum_{i=1}^{K(S)} f_i(S) = 1$$

To quantify the degree to which sensor S is stable in its reading, we apply the notion of information entropy. The entropy of the values of a sensor S, denoted by $VEntropy(S)$, is defined by

$$VEntropy(S) = -\sum_{i=1}^{K} f_i(S) \cdot \log f_i(S)$$

where $VEntropy(S)$ is maximum when all ranges of values of S are equally likely (i.e., when S changes value often). It is minimum when the values of S have all been in one range $R_i(S)$, thus $f_i(S) = 1$ (for some i, $1 \leq i \leq K(S)$). It can easily be shown that $0 \leq VEntropy(S) \leq \log K$. We are now ready to define the average value informativeness of sensor S, denoted by $VInform(S)$, to be

$$VInform(S) = 1 - \frac{VEntropy(S)}{\log K(S)}$$

where $VInform(S)$ takes on values between 0 and 1. A value of 1 indicates that S normally rarely changes its value, while a value of 0 indicates that S's value is equally likely to be in any of its ranges.

On the other hand, the quantity

$$VUnusual(S) = 1 - f_i(S)$$

gives the unusualness of sensor S's value being in the i-th bin. $VUnusual(S)$ is computed each time S reports a value, and the i used is the index of the bin containing the reported value. This measure can assign the same degree of unusualness in fundamentally different situations. For instance, it does not distinguish between a value having a probability of $\frac{1}{K}$ occurring when all other values have an equal probability of $\frac{1}{K}$ each, and a value with probability $\frac{1}{K}$ when only one other value has probability $(1 - \frac{1}{K})$ with the remaining values having probability 0. In the first case, the value is just as likely as any other. In the second case, the interesting event is that the most likely value did not occur. To make this distinction we combine the unusualness and value entropy measures to obtain the ***surprise*** score:

$$Surprise(S) = VInform(S) \cdot VUnusual(S).$$

This measure takes on the maximum value of 1 when one bin in the histogram has probability one and the sensor registers a value in another bin. It has a minimum value of zero when all bins in the histogram are equally likely.

Accounting for Alarm Thresholds Alarm thresholds for sensors, indexed by operating mode, typically are established through an offline analysis of the design of NASA space systems. SELMON makes use of alarm threshold information in the following way: A sensor whose value traverses the safety threshold is said to go into a state of alarm. The predicate *In_Alarm*(S) captures this notion:

$$In\_Alarm(S) = \begin{cases} 1 & \text{if } S \text{ is outside its safety range} \\ 0 & \text{if } S \text{ is within its safety range} \end{cases}$$

We compute the value of an alarm score for S as follows:

$$ALScore(S) = In\_Alarm(S) \cdot [1 + Trav(S)] \,.$$

where $Trav(S)$ is the proportion of the alarm range traversed.

We consider alarms as interesting events whose importance decreases with time. Thus a sensor that persists in alarm state for prolonged periods of time should gradually fade from our attention. To achieve this we add an exponential decay factor. Let $t_A(S)$ be the time at which sensor S last entered into alarm. At any time t, the alarm score is computed as follows:

$$Alarm\_Score(S) = \frac{1}{2} ALScore(S) e^{-\beta(t - t_A(S))}$$

where $\beta > 0$ is the time decay constant. β is chosen small so the decay will not be too fast; typically $\beta \leq 0.1$/second.

Given the recent values of S, one may conduct a simple form of trend analysis to decide whether or not sensor S is anticipated to be in alarm soon. The measure *Predict_Alarm*(S) is a curve-fitting prediction of when the sensor will enter alarm. This measure has a minimum of 1 and a maximum of infinity if the curve fit indicates that the sensor will never enter alarm. If the sensor is currently in alarm, *Predict_Alarm*(S) measures when the sensor is predicted to leave alarm. This measure is used to compute a score *Anticipate Alarm* as follows:

$$Anticipate\_Alarm(S) = \begin{cases} 1/Predict\_Alarm \\ 1 - 1/Predict\_Alarm \end{cases}$$

The first case applies when S is within its safety range. The second case applies when S is outside its safety range.

Thus, if S is currently not in alarm, *Anticipate_Alarm* will be at its maximum of 1 when *Predict_Alarm* predicts the sensor will enter an alarm range immediately. If S is currently not in alarm, *Anticipate_Alarm* will be at its minimum of 0 when *Predict_Alarm* predicts the sensor will never enter alarm. If S is currently in alarm , *Anticipate_Alarm* will be at its maximum of 1 when *Predict_Alarm* predicts the sensor will never leave the alarm range. If S is currently in alarm, *Anticipate_Alarm* will be at its minimum of 0 when *Predict_Alarm* predicts the sensor will immediately leave alarm.

Quantifying Value Change A change in the value of a sensor is considered to be an event of interest. The surprise measure described above measures the degree of interestingness of a sensor taking on a certain value. Another aspect of sensor behavior to measure is the most recent change in value of the sensor that brought it to its current reading. However, absolute change magnitude is not interesting in and of itself. What is interesting is the probability of the most recent change taking place. Hence we need a scheme for normalizing the absolute change in value of a sensor.

The scheme we use assigns a score to each change in the value of a sensor that is an estimate of the proportion of all previous value changes for that sensor that had value changes strictly less than the change under consideration. Suppose we get a change in value of the sensor equal to Δ. Furthermore, suppose that 60% of the previous value changes for this sensor in the current operating mode have been less than Δ. In this case, we assign a score of 0.6 to the change Δ. Changes with magnitude greater than Δ will get higher scores.

This scheme requires that we keep track of a sorted sequence of all value changes of each sensor. This is neither feasible nor necessary. An approximation of this value can be obtained by keeping a constant number of values, say W, in a sorted sequence. Let the total number of changes in the values of a sensor so far be $C(S)$. Rather than storing all $C(S)$ values, we store only $W < C(S)$ values. With the arrival of a new change in value for sensor S, we increment the count of changes $C(S)$ and then we decide whether to replace one of the W values we are storing or simply ignore the current value change. The decision criterion is to generate a random number in $[0, 1]$ according to a uniform distribution, and replace one of the W values if and only if that random number is less than $\frac{W}{C(S)}$. It can be proven that this algorithm is equivalent to one that stores all $C(S)$ values, randomly samples W of them, and returns as score the proportion of the W elements that have value less than the change under consideration.

We call this score the *percentile value change* score. It is used to assign a normalized score in the range $[0, 1]$ for each change of value that occurs in each sensor. By definition, this score is maximum when the change is the maximum change of value seen so far for a particular sensor. It is minimum when no change occurs in the value of a sensor.

Model-Based Measures

SELMON also uses a model of the monitored system to determine sensor importance. This model is used to compute three scores: deviation, sensitivity, and cascading alarms. This section describes how each of these scores is computed.

Deviation The deviation measure uses a model of the monitored system to make predictions of expected current sensor readings. The concept of the deviation score is that sensor readings deviating significantly from the predicted values are anomalous and should be reported to the operator.

The deviation score is computed in the following manner. First, the raw deviation is computed as the difference between the predicted and observed sensor scores. This raw deviation is entered into a normalization process identical to that used for the value change score, and the resultant score in the range [0,1] is the overall deviation score.

Causal Analysis The SELMON system also uses the causal model of the monitored system to reason about future effects of current quantity changes. These future effects are considered in two causal-based measures. First, *sensitivity* measures the effect of predicted changes in quantities on the overall state of the system. This is done by projecting each predicted change in a quantity individually forward as a perturbation of the system, and measuring the overall change in the system. Those currently occurring changes which have a greater effect upon the future state of the system are likely to be more important and thus receive high scores to be displayed to the operators. The second causal reasoning measure is *cascading alarms*, which measures the potential for observed changes to result in rapidly developing alarm sequences. The cascading alarms measure uses the same perturbation analysis used in the sensitivity analysis and measures the number of alarms triggered and how quickly alarms occur. Those predicted changes which are expected to trigger large numbers of alarms are scored highly and thus will be selected to be displayed to operators.

Sensitivity Analysis Sensitivity analysis measures the sensitivity of other quantities in the monitored system to changes in each quantity in the model. This is performed as follows. Beginning with a simulation of the system in its current state and time $T_{current}$, simulate forward one timestep (i.e. until the next time sensors are expected to be polled). For each quantity Q, choose ΔQ_{pred} as the current 50th percentile value change recorded for the given sensor.

Then, for each quantity Q, run a simulation beginning again with the current system state, perturbing Q by ΔQ_{pred}, propagating this change to other quantities in *All_Quantities* (the set of all quantities in the model) as dictated by the model. For each such changed quantity Q' in *All_Quantities*, for each time *time*1 that the quantity changes during the simulation, collect a sensitivity score proportional to the amount of change in Q' normalized to the size of the nominal range of the sensor but also modified by a decreasing function of *time*1. This calculation captures the characteristic that delayed and less direct effects are more likely to be controllable and less likely to occur. Thus, a change which affected a quantity Q' but occurred slowly is considered less important. This simulation proceeds for a predetermined amount of simulated time. Then, for each changed quantity Q', take the maximum of the collected *change_scores* for that quantity. The sensitivity score for Q is the sum of these maximums for all the Q's. Thus, for each quantity Q, a simulated change produces a set of *change_scores* for each other quantity in the model. The sensitivity score for Q is the sum of the respective maximums of each of these sets. If there are no changes to a quantity, this set is empty and the quantity receives a zero score.

A background sensitivity score is subtracted from the sensitivity score for Q, computed by measuring the sensitivity score via simulation with no perturbation of the system.

Cascading Alarms Analysis Cascading alarms analysis measures the potential for change in a single quantity to cause a large number of alarm states to occur, thus causing information overload and confusion for operators. In the cascading alarms score, the same simulation used in the sensitivity score computation is used to also determine the number of alarms triggered by the observed change. In the cascading alarms score, for each quantity Q, the number of alarms triggered by a perturbation of Q by ΔQ_{pred} is computed.

The alarm count is then normalised for the total number of possible alarms and the weight of each alarm state triggered is also decreased as a function of the time delay from the initial change event to the alarm. This has the effect of focussing this measure on quickly developing cascading alarm sequences which are the most difficult to interpret and diagnose. Finally, the cascading alarms score is normalized by subtracting the background cascading alarms score. This background score is simply the cascading alarms score for no perturbation.

Computing a Total Sensor Score

We use the *surprise* score to modulate the percentile *value change* associated with a sensor. This accounts for the unusualness of a sensor value as well as the change in the sensor value that brought it to its current reading. The percentile value change score is also used to modulate the scores obtained by the causal analysis of the system: the *sensitivity* score and the *cascading alarms* score. These are modulated by the percentile *value change* because they are computed based on an analysis of the effect of a perturbation in the value of the sensor on the overall system. The remainder of the score combinations are simple sums. See Figure 2.

Application Domain

Our application domain is the hardware testbed of the water side of the Environmental Control and Life Support System (ECLSS) for Space Station Freedom. The water side of ECLSS consists of three principal systems: Multifiltration (MF), Vapor Compression and Distillation (VCD), and the Volatile Removal Assembly (VRA). Using a combination of analysis of system description documents, consultation with testbed engineers, and actual hardware testbed data, we have constructed models of all three of these subsystems. Each subsystem model contains 30-50 quantities and 15-30

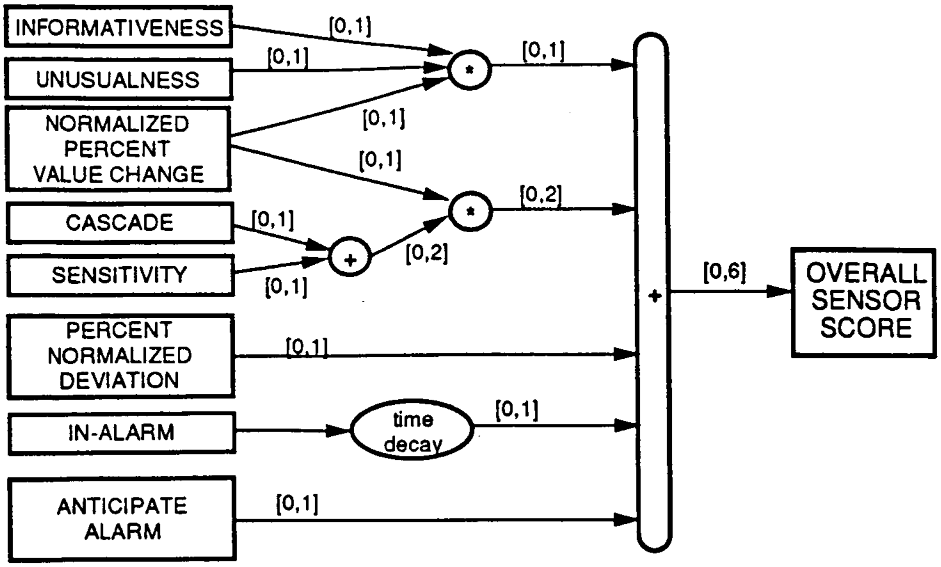

Figure 2: SELMON Sensor Scoring Algorithm.

mechanisms. Work in elaborating fault models is ongoing. This model has been validated by comparison against actual data from the subsystem testbed undergoing evaluation at the Marshall Space Flight Center (MSFC) in Huntsville, Alabama. We are also in the process of extending our model to cover the ECLSS air side subsystems.

Performance Evaluation

The output of the SELMON algorithms is dynamically computed each time the sensors are polled. SELMON produces a total ordering by importance on the set of sensors, and a window size which determines how many sensor data are presented to the operator. In order to assess whether SELMON is usefully focusing operator attention, we are comparing sensor subsets selected by SELMON to critical sensor subsets specified by domain experts as useful in understanding episodes of anomalous behavior in actual historical data from ECLSS testbed operations.

In one experiment, we asked whether or not SELMON was suppressing sensor data deemed critical by a domain expert. For this experiment, we separated the performance of the window sizing algorithm from the sensor scoring algorithm by choosing a constant window size. The specific question posed was how often did SELMON place a "critical" sensor in the top half of the sensor ordering. For a sensor set of cardinality 13, we defined the top half to be the first seven slots in the total sensor ordering. Thus the performance of a random sensor selection algorithm would be expected to be about 46.2%. Table I shows the results of this experiment. The first column identifies one of the episodes specified by the domain expert. The second column shows the number of timesteps in the episode in which the given sensor was deemed critical. The third column shows the overall SELMON "hit" rate for that episode: the number of times SELMON placed the given sensor in the top half of the sensor ordering.

EPISODE	# of timesteps	Hit Rate (%)
kc01.1	710	81.4
kf01.1	3	100
kf01.2	7	100
kf01.3	7	100
kf01.4	2	100
kf01.5	2	100
kf01.6	2	100
kf01.7	2	100
kf01.8	2	100
kf01.9	7	100
kf01.10	4	50.0
kp01.1	40	47.5
kp02.1	40	47.5
kp03.1	40	62.5
kp01.2	71	98.6
kp02.2	71	100
kp03.2	71	100
kt01.1	27	100
kt02.1	9	88.9
kt02.2	332	100
kt04.1	25	100
All	1512	87.1

Table I: SELMON performance at selecting critical sensor data.

These results suggest that SELMON performs at much better than random at replicating the attention focussing of one domain expert identifying episodes of anomalous behavior for the ECLSS testbed. SELMON's performance is not yet at the level which could support an operational capability for real-time monitoring assistance. A more detailed analysis is ongoing to determine why SELMON performed poorly in some episodes and to examine the performance for individual sensor importance measures.

SELMON is intended to assist operators in efficient anomaly detection – the first step towards diagnosis. Another planned experiment will investigate how sensor selection supports diagnostic reasoning:

In addition to the ECLSS subsystem models which describe nominal behavior, a number of ECLSS fault models are being developed. After implementing a diagnostic reasoning algorithm, we will determine how this algorithm performs at correctly diagnosing faults from behavior traces resulting from simulation of these fault models. We will then test the performance of the diagnostic reasoning algorithm when it is given only SELMON-selected sensor data. Finally, we will test the performance of this algorithm when it is given the same number of sensor data randomly selected. Some degradation of performance is expected in the diagnostic reasoning algorithm using SELMON-selected data. A measure of success will be a significantly greater loss of performance with randomly selected data. A final caveat is that this experiment may only indirectly shed light on the ability of SELMON to support *human* troubleshooting activity.

Discussion

NASA mission operators are trained to interpret raw telemetry to create a mental model of the state of a spacecraft or spacecraft subsystem. SELMON is intended to focus operator attention on the most important sensor data. If SELMON does nothing more, it may be construed to be simply and only providing operators with less raw data to interpret, and thus may be considered to be a step in the wrong direction.

Accordingly, we recognize that an important component of the SELMON approach is the ability to provide explanations or interpretations of why a particular sensor has been placed in the monitoring window and is worthy of operator attention. Future work in the SELMON project will be oriented towards complementing focus of attention and anomaly detection capabilities with model-based interpretation capabilities.

In related work, we are also investigating the problem of sensor placement during design, using both monitorability [Chien et al 91a] and diagnosability [Chien et al 91b] criteria.

Summary

We are developing techniques to support real-time monitoring through sensor selection, the moment to moment focusing of attention on a subset of the available sensor data. Sensor selection is based on a set of importance criteria which draw on concepts from model-based reasoning and information theory. Although the SELMON project is currently targeted towards focus of human operator attention, the techniques may also support focus of attention in an autonomous monitoring and control system.

Acknowledgements

Others who have worked recently on the SELMON project include Leonard Charest and Nicolas Rouquette. We would like to thank Jay Wyatt of the Marshall Space Flight Center for many informative discussions regarding the operation of the ECLSS system.

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

[Chien et al 92] S. A. Chien, R. J. Doyle, and U. M. Fayyad, "Focusing Attention in Real-Time Systems Monitoring," *AAAI Spring Symposium on Selective Perception*, Stanford, March 1992.

[Chien et al 91a] S. A Chien, R. J. Doyle, and L. S. Homem de Mello, "A Model-based Reasoning Approach to Sensor Placement for Monitorability," *3rd AAAI Model-Based Reasoning Workshop*, Anaheim, July 1991.

[Chien et al 91b] S. A Chien, R. J. Doyle, and N. F. Rouquette, "A Model-based Reasoning Approach to Sensor Placement for Diagnosability," *2nd International Workshop on the Principles of Diagnosis*, Milan, October 1991.

[Doyle et al 92] R. J. Doyle, D. Berleant, L. K. Charest, Jr., U. M. Fayyad, L. S. Homem de Mello, H. J. Porta, M. D. Wiesmeyer, "Sensor Selection in Complex Systems Monitoring Using Information Quantification and Causal Reasoning," in *Recent Advanced in Qualitative Physics*, B. Faltings and P. Struss (eds.), MIT Press, *to appear*.

[Doyle and Fayyad 91] R. J. Doyle and U. M. Fayyad, "Sensor Selection Techniques in Device Monitoring," *2nd Conference on AI, Simulation and Planning in High Autonomy Systems*, Cocoa Beach, April 1991.

[Doyle et al 86] R. J. Doyle, D. J. Atkinson, and R. S. Doshi, "Generating Perception Requests and Expectations to Verify the Execution of Plans," *5th National Conference on Artificial Intelligence*, Philadelphia, August 1986.