Situation Management in the Link Monitor and Control Operator Assistant (LMCOA)

Randall W. Hill, Jr. and Lorrine F. Lee

Jet Propulsion Laboratory
California Institute of Technology
4800 Oak Grove Drive
Pasadena, CA, 91109-8099, USA
MS 525-3631 & 525-3660

ABSTRACT

This paper describes a knowledge-based system called the Situation Manager that was developed for the Link Monitor and Control Operator Assistant (LMCOA) at the Jet Propulsion Laboratory. This system was developed in response to a number of deficiencies that were identified in an earlier version of the LMCOA: (1) the need to close the control loop between sending a directive and knowing when its execution is complete (versus just closing the communications loop), (2) the need to recognize an anomaly and alert the operator when a directive is rejected or a link device fails, and (3) the need to suggest ways to work around an anomaly, provided that it is recognizable. In response to these needs, the Situation Manager has been designed to provide the LMCOA with three basic capabilities: situation assessment, anomaly diagnosis, and recovery from commonly occurring problems.

Key Words: Monitor and control, knowledge-based system, operator assistant, Deep Space Network, automation

1. INTRODUCTION

The goal of the Link Monitor and Control Operator Assistant (LMCOA) is to increase the availability of the Deep Space Network (DSN) by reducing the amount of time it takes to configure, calibrate and operate the communications links used to support deep space communications missions (Ref. 1). The LMCOA seeks to achieve this goal by decreasing the cognitive and motor load that is currently placed on LMC operators (Ref. 2,4). This paper describes the Situation Manager, the knowledge-based system at the heart of the LMCOA. The Situation Manager supports the LMCOA in the following ways: (1) it monitors device states, (2) it evaluates the appropriateness of control actions with respect to the device situation, and (3) it diagnoses and recovers from simple situational anomalies.

To illustrate the need for an Operator Assistant, consider the following: during a typical pass an LMC Operator interacts with five different subsystems consisting of fifty or more devices, where each device has multiple attribute-values. In all, the operator monitors 250 or more unique data elements on subsystem displays and reads 500 or more event notice messages indicating significant state changes in subsystem devices. While performing the task, the operator issues 200 or more directives to control subsystem devices and gain access to subsystem data, and for each directive issued a response is returned that indicates whether it was accepted or rejected by the subsystem. As a consequence of the current system design, a large portion of the operations time is spent just configuring and calibrating the communications link, which is becoming increasingly more unacceptable in light of the increased demand for DSN availability for mission support.

In the next section we describe the Device Model, which is the Situation Manager's internal representation of the communications link being monitored. Following the discussion of the Device Model, we discuss how control actions are modeled in the Situation Manager. The Action Model was derived through extensive interviews with LMC operators and engineers, and it is encapsulated in a representation called the Temporal Dependency Network (TDN) (Ref. 3). Finally, we describe how the Situation Manager uses the knowledge about devices and control actions to support monitor and control activities being implemented in the LMCOA.

2. DEVICE MODEL

One of the primary tasks performed by LMC operators is to monitor the state of the devices in the communications link. The Situation Manager performs this function for the LMCOA

Device	Attribute	Value
RECORDING-DEVICE	NAME	LD0
	SELECTED?	YES
	MODE	RECORD

Table 1. Device Model Representation

by keeping an attribute-value model of all of the relevant link devices. An attribute is a property of the device that can be monitored or evaluated. An attribute-value represents the actual state of a particular device property.

2.1 Representation

A device is any entity in the communications link that has an observable state. Thus, the Device Model includes physical entities such as the Phase Calibration Generator, the Digital Tone Extractor and the Recording Devices, and it also includes software (e.g. the Narrow Channel Bandwidth (NCB) Program and the Predicts File). Each device has a name and a variable number of attributes and values. A typical device representation is shown in Table 1.

2.2 Sources of information

The LMCOA routes data about the state of the link devices to the Situation Manager's Device Model from several different sources: event notice messages, subsystem monitor data, directive responses and user input. Event notice messages describe significant state changes in the devices being monitored and they are always sent to update the Device Model. Likewise, directive responses are always routed directly to the Situation Manager, though they do not always provide useful state information. Monitor data, on the other hand, is not sent directly to the Device Model. Rather, the Situation Manager requests this data from the LMCOA's monitor database only when it is needed. This policy avoids the problem of inundating the Situation Manager with frequent updates which often reflect no change in value. The Situation Manager may also request user input for device state information when other information sources fail.

3. ACTION MODEL

Whereas the Device Model is used to represent and monitor events in the communications link, the Action Model represents the knowledge of when to send directives and how to know whether they have had their desired effects. The Action Model is represented in a framework called the TDN that explicitly encodes knowledge about procedures, directives and their situational dependencies, thereby providing a pass-specific, integrated, end-to-end operational sequence of action.

We determined the need for the TDN by studying the behavior of several different expert LMC operators performing a particular type of pass called Very Long Baseline Interferometry Delta Differential Oneway Ranging (VLBI Delta DOR). We found that there was much more to their operational expertise than merely following the procedure manuals. Though explicit directive sequences are given in the procedure manuals, an expert operator typically brings additional knowledge to bear on the task. Specifically, the experts have an implicit knowledge about *directive preconditions and postconditions*. In other words, the operator knows a directive will only have its desired effect when the devices are in a particular state (i.e., preconditions), otherwise the directive will be rejected, or worse, it will have a deleterious effect on the target device. Once a directive is issued, the operator knows what effects to expect in the devices (i.e. postconditions), and if the effects are not observed then something is wrong. Thus, the expert LMC operator combines situational awareness of the communications link devices with knowledge about directive preconditions and postconditions.

In addition to having knowledge of directive preconditions and postconditions, expert LMC operators also have an implicit knowledge of how to order the procedures they perform. The procedure manuals are typically subsystem oriented, but since the task entails interacting with many subsystems the LMC operator must choose an order in which to execute the procedures. There is no overall description of the operational sequence which optimally depicts how to execute the procedures, so we find that the same task goals can be achieved by different orderings among procedures and directives. Moreover, we find that LMC operators attempt to optimize the overall task by interleaving procedures for different subsystems whenever possible.

3.1 TDN Blocks as Default Procedures

One of the goals of the TDN is to optimize task

Label	Directive	Device	Attribute	Value
PRE1	NLOAD JK	VLBI-PREDICT-DATA	RECEIVED?	YES
PRE2	NLOAD JK	VLBI-PREDICT-DATA	NAME	JK
PRE3	NLOAD JK	NCB-PROGRAM	MODE	IDLE
POST1	NLOAD JK	VLBI-PREDICT-DATA	LOADED?	YES

Table 2. Representation of Directive Preconditions and Postconditions

performance by interleaving procedure execution. One of the ways that this is achieved is by using the TDN block, which is the primary unit of organization in the TDN. A TDN block contains a sequence of directives, where there are no interdependencies among the directives within the block. The execution order among TDN Blocks is defined by a pairwise precedence relationship. This precedence relationship is transitive and creates a partial order for the directives in the affected blocks. If no order is specified between two blocks, either explicitly or implicitly by transitivity, procedural interleaving is achieved by executing the independent partial orders of directives in parallel. Otherwise, a block of directives must successfully execute before its successor block is allowed to begin its execution.

3.2 Directive Preconditions and Postconditions

Once a block of directives is eligible for execution (i.e. when its successor block is done), the situation must be evaluated to determine whether the current block's directives can be safely issued. The situation is evaluated using directive preconditions and postconditions, which represent a situational relationship between a directive and one or more devices. Each directive may have multiple preconditions and postconditions. For instance, in Table 2 the "NLOAD JK" directive has a precondition labeled PRE1 that says the predicts file must be present (i.e., RECEIVED? YES) in order for the directive to be accepted. In addition, the predicts file must have the name "JK", since this is the parameter specified by the NLOAD command. Finally, NLOAD also has the less obvious precondition, labeled PRE3, which states that the Narrow Channel Bandwidth Program must be in the IDLE mode (rather than in the RUN mode). The postcondition for this directive, labeled POST1, states that the subsystem must give an indication that the predicts file was successfully loaded once the directive has been issued.

4. SITUATION MANAGER

The Situation Manager provides cognitive support to the LMCOA by monitoring and evaluating devices in the communications link and by making situated control decisions for the TDN Execution Manager. To accomplish this the Situation Manager uses two kinds of knowledge: mission-specific and operational. Mission-specific knowledge is represented declaratively and it includes the Device and Action Models previously described. Operational knowledge is represented using productions that match against the mission-specific knowledge and perform different tasks. In this section we discuss how the mission-specific knowledge is used by the Situation Manager's operational knowledge to perform the following functions: (1) process input messages, (2) update the Device and Action models, (3) evaluate the situation, (4) diagnose anomalies, and (5) recover from anomalies.

4.1 Flow of Control and Data

The Situation Manager interacts extensively with other LMCOA modules (Figure 1). In this section we give a detailed account of these interactions in order to provide a context for describing the individual functions within the Situation Manager.

The Situation Manager loads mission-specific Device and Action models during the initialization phase. Once a mission begins, the TDN Execution Manager analyzes the TDN and chooses a set of TDN blocks to execute based upon the precedence relations among the blocks. Prior to issuing the directives contained in the blocks, however, the TDN Execution Manager consults with the Situation Manager about whether all of the preconditions are satisfied for the directives in the selected blocks. This consultation takes place in two phases. First, the TDN Execution Manager sends a copy of each of the parameterized directives to the Situation Manager, which uses the parameter information to update the preconditions in its Action Model. Second, it requests the Situation Manager to evaluate the preconditions for each of the TDN blocks. If the preconditions for a block of directives are all satisfied, the Situation Manager gives permission for the

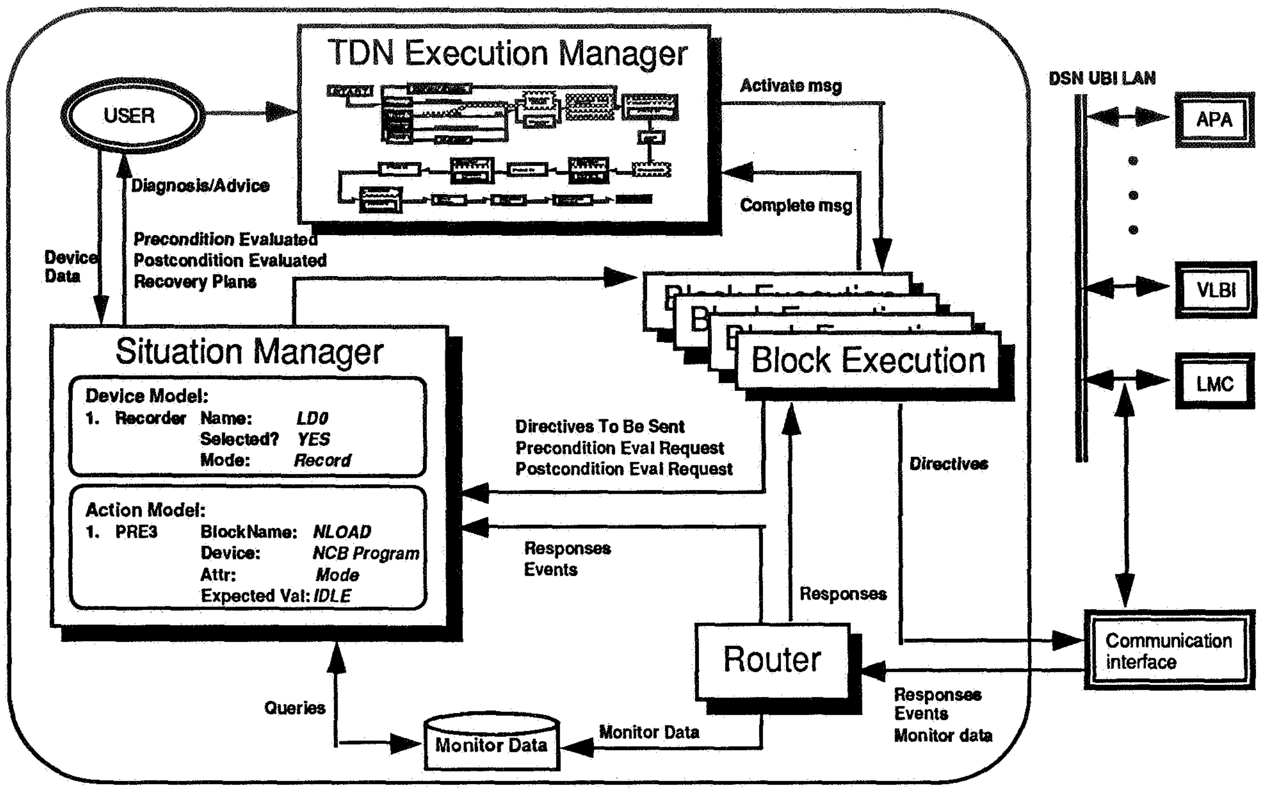

Figure 1. Data and Control Flow in LMCOA

block to be executed, and the directives are issued to the subsystems in the communications link. If any preconditions are not satisfied, the Situation Manager suspends the block's execution and sends a message to the user interface indicating that there is a problem with a specific directive associated with a particular device attribute. The message indicates both the actual and the expected device attribute values, and the user is expected to respond by either fixing the problem that was noted in the message, or by providing the current actual value for the Device Model. The latter action assumes that the user has access to information that is more current than the Situation Manager's Device Model. Once the Situation Manager sees a response to the precondition error message, it re-evaluates the block's preconditions. If the same precondition is still unsatisfied, the user has the option of overriding the precondition so that the block's directives will be issued in spite of the Situation Manager's warnings. Once a block's directives have been issued, the TDN Execution Manager requests the Situation Manager to evaluate whether the block's postconditions are satisfied (i.e. check whether all the block's directives executed as expected). The postcondition evaluation proceeds in much the same way as the precondition evaluation.

4.2 Processing Input Messages

Each time the Situation Manager is invoked it first enters a phase where it processes input messages. It uses knowledge base rules to recognize the different types of messages and parses them into forms usable by subsequent phases. The message types handled by the system include event notice messages, directives, directive responses, directive rejections, user-responses, user-override-responses, and requests for precondition or postcondition evaluation.

4.3 Updating Device and Action Models

The Device and Action models are initialized with a set of default values from their mission-specific knowledge bases. The models change rapidly over time in response to changes in the communications link and adjustments in the procedures. The Device Model is affected primarily by event notice messages and directive responses. Each event notice message type has a specialized rule that reads and interprets the

Device	Attribute	Old Value	New Value
VLBI-PREDICT-DATA	RECEIVED?	NO	YES
	NAME	NULL	VN

Table 3. Device Model Change Caused by an Event Notice Message

Label	Directive	Device	Attribute	Old Value	New Value
PRE2	NLOAD JK	VLBI-PREDICT-DATA	NAME	JK	VN

Table 4. Modification to a Precondition via Directive Parameterization

message string and subsequently updates the appropriate device in the model. To illustrate this point, consider a message that is generated after the predicts set has been received by one of the subsystems: PRED SET VN RECEIVED (008). This message causes the modifications to the Device Model shown in Table 3.

Similarly, when the TDN Execution Manager sends a directive to the Situation Manager the Action Model is modified to reflect the parameters used with the directive. For instance, if the directive "NLOAD VN" was sent to the Situation Manager, the precondition we previously defined in the Action Model would be modified to reflect the "VN" parameter as shown in Table 4. Thus, by sending the Situation Manager a copy of the parameterized directive, the affected precondition for the directive is updated with the correct parameter value.

4.4 Evaluating Preconditions and Postconditions

When asked to evaluate the preconditions (or postconditions) of a TDN block, the Situation Manager retrieves all of the block's directives and their associated preconditions. The precondition specifies an expected value for a particular device attribute. Thus the evaluation consists of comparing the expected device attribute value with an actual device attribute value. If the two values are the same, then the evaluator records that the precondition is satisfied and continues the evaluation with the next precondition. If the two values are not the same, then it immediately diagnoses the problem and invokes a simple recovery scheme that will be described below. Control is returned to the evaluation procedure once the asynchronous recovery scheme is invoked, and the evaluation continues until all of the preconditions have been processed. If there are one or more unsatisfied preconditions, the block will be suspended and re-evaluated once the user has taken corrective action.

4.5. Diagnosing anomalies

There are three basic types of anomalies: (1) pre/postcondition failures, (2) directive rejections, and (3) event notice alarm messages. Precondition and postcondition failures occur when the actual device state is not equal to the expected state for a particular directive. The diagnosis in this case consists of the following steps: (1) isolate the device attribute for which the evaluation failed, (2) verify that the associated value in the device model is current by polling the monitor database, and (3) if steps 1 and 2 still indicate an unsatisfied pre/postcondition, notify the user with an asynchronous diagnostic message containing the device name, attribute, and its expected and actual values. The other two types of anomalies, directive rejections and event notice message alarms, have standardized explanations associated with them that are retrieved by the knowledge base's diagnostic rules.

4.6. Recovering from anomalies

The Situation Manager helps the operator to recover from pre/postcondition failures by providing diagnostic information about the failure in a dialogue window. The message contains the TDN block name, directive name, device name, attribute, and the expected and actual values. Recall that the actual value comes from the Device Model and may not reflect reality. Hence, recovery begins when the operator compares the Device Model actual value with reality for the indicated device attribute. If the Device Model's actual value does not match reality, the operator updates the value in the dialogue box. If the Device Model's actual value is accurate, then the operator indicates this is so in the dialogue box. In this way, the Situation Manager insures that the Device Model is current and re-evaluates the pre/postconditions of the suspended block. If preconditions are still not satisfied after being evaluated a second time, the user can override the precondition, thereby breaking out of a potential loop. Otherwise, if the operator chooses not to override the unsatisfied precondition the cycle of diagnosis and recovery will be repeated, and the operator can attempt to resolve the disparity by issuing separate commands to the subsystem in question to place the device in the expected state.

The Situation Manager helps the operator to recover from directive rejections and event alarms by providing a narrative description of the directives that should be issued in order to solve the problem. Future work will focus on how to dynamically construct a recovery block of directives to deal with all three anomaly types. Though the current system assists the operator in recovery by providing diagnostic information and other descriptive narrative, it does not dynamically compose directive sequences to solve problems.

5. SUMMARY

In attempting to improve the operability of the LMC system, we seek to reduce the cognitive load that is currently placed on LMC operators. This is accomplished by providing automated support to both the monitor and the control activities performed by the operator. The LMC Operator Assistant Situation Manager maintains a dynamic model of communications link devices and uses this model to evaluate the preconditions and postconditions of control actions. In cases where the preconditions or postconditions are not satisfied, the Situation Manager provides specific feedback to the operator about why an action should not be taken or why it failed.

6. ACKNOWLEDGEMENTS

The work described in this paper was carried out by the Jet Propulsion Laboratory, California Institute of Technology, under contract with the National Aeronautics and Space Administration. The other members of the LMCOA team are Sanguan Chow, Lynne Cooper, Kristina Fayyad, Kathy Sturdevant and Juan Urista. We would like to acknowledge the contributions of Pam Wolken, Terry Dow, Dave Girdner, Elmain Martinez and the many operations personnel at Goldstone Deep Space Communications Complex who have shared their time and expertise with us in this effort.

7. REFERENCES

1. Cooper, Lynne P., Rajiv Desai and Elmain Martinez, "Operator Assistant to Support Deep Space Network Link Monitor and Control," SOAR Symposium, Houston, TX, 1991.

2. Cooper, Lynne P., "Operations Automation Using Temporal Dependency Networks," Technology 2001, San Jose, CA, December 3-5, 1991.

3. Fayyad, Kristina E., and Lynne P. Cooper, "Representing Operations Procedures Using Temporal Dependency Networks", SpaceOPS 92: Second International Symposium on Ground Data Systems for Space Mission Operations, Pasadena, CA, 1992.

4. Lee, Lorrine and Randall W. Hill, Jr., "Process Control and Recovery in the Link Monitor and Control Operator Assistant," SOAR Symposium, Houston, TX, 1992.