N90-27306

An Application of Design Knowledge Captured from Multiple Sources

Preston A. Cox
John H. Forbes
Lockheed Missiles and Space Company
Org. 62-81, Bldg 579
1111 Lockheed Way
Sunnyvale, California 94089-3504

Abstract

The Hubble Space Telescope Operational Readiness Expert Safemode Investigation System (HSTORESIS) is a reusable knowledge base shell used to demonstrate the integration and application of design knowledge captured from multiple technical domains. The design of HSTORESIS is based on a partitioning of knowledge to maximize the potential for reuse of certain types of knowledge.

Introduction

The Hubble Space Telescope Operational Readiness Expert Safemode Investigation System (HSTORESIS) is a knowledge based system which demonstrates the integration and application of design knowledge captured from multiple technical domains. The domains of interest are the electrical power and pointing control systems of the Hubble Space Telescope (HST). The types of design and engineering knowledge contained in HSTORESIS pertain to the analysis and resolution of system anomalies known as safemode events.

HSTORESIS is motivated by the HST Design/Engineering Knowledge-base (HSTDEK) project. The primary goals of the HSTDEK project are to enable major NASA projects to capture design and engineering expertise and to support the use of the captured knowledge in multiple applications [2]. HSTORESIS addresses these goals by providing a reusable knowledge base shell which can access a variety of device models and rule bases to allow a user to solve a variety of problems.

The following sections discuss some key technical issues addressed in HSTORESIS and describe major features of HSTORESIS.

Knowledge Partitioning

It has been said that one important approach to managing the computational cost of causal reasoning is structural abstraction [3]. In this spirit, the design of HSTORESIS is based on a partitioning of the knowledge typically contained in a knowledge based system. The knowledge partitioning helps manage the computational cost inherent in rule based systems and increases the opportunities for knowledge reuse.

It is not practical or possible to procedurally define all of the behaviors of a complex device even though, from an engineering perspective, each subcomponent may be precisely defined. In order to reason about a device as complex as the HST, a system must include both procedural or algorithmic knowledge and heuristic or partial knowledge. Traditionally, applications using the production system approach tend to merge both types of knowledge into one rule base.

Merging procedural and heuristic knowledge contributes to system brittleness and reduces the opportunity for knowledge reuse. To overcome these problems, HSTORESIS provides the hooks for the knowledge engineer to partition a knowledge base into procedurally oriented device models and heuristically oriented production system rules. This knowledge base partitioning allows more than one set of heuristics, or production rules, to be applied to an HST component or subsystem. This increases the potential for knowledge reuse.

For simple systems, the encoding of the procedural knowledge in device models is often sufficient to describe the system's behavior. However, because of its complexity, the HST is not a fully described system. Some of its behaviors cannot be procedurally described. For example, a design engineer might know from experience that if a reaction wheel spins at 2,200 rpm for more than two minutes the rotor bearings will experience excessive wear. The engineer might therefore recommend that maneuvers be avoided that would cause the reaction wheel to over spin for more than one and a half minutes.

The important distinction is that heuristic knowledge is only approximate and is subject to different interpretations in different situations. For example, how much bearing wear can be tolerated might depend on the importance of making a particular maneuver or the nearness in time to a service interval. In contrast, the calculation of angular momentum or the mass of the reaction wheel is a fixed characteristic of the device.

The reason for making this distinction is that the procedural knowledge has a greater potential for reuse. This reuse can be achieved in two ways:

- By having more than one set of heuristic rules reason over the same device model, and

- By combining device models to create new models. (This type of reuse, composite models, is possible because both single and multiple inheritance are supported by the object oriented programming tool used by HSTORESIS.)

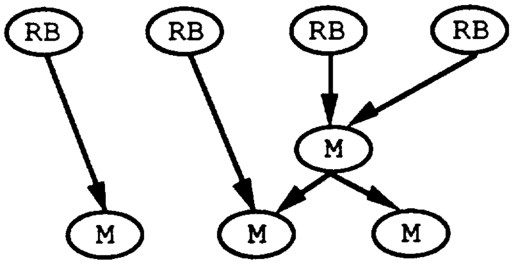

Figure 1. Example of Reuse in HSTORESIS

Figure 1 illustrates both types of reuse. In one case, two rule bases (the bubbles labelled RB) reason over one model. The model itself is a composite of two other models.

In contrast to the procedural knowledge, the heuristic knowledge is subject to more change over time. For this reason, the proposed partitioning will make knowledge bases easier for the knowledge engineer to maintain and modify.

Reusable Interface

An analyst interacts with the knowledge contained in HSTORESIS through the Interface Management System. The basic script that the analyst follows to define a problem for analysis is suggested by Figure 2. The analyst selects a device model from a menu of all models known to HSTORESIS. The analyst also provides a time interval over which the device model is to be analyzed.

The Interface Management System retrieves from the selected device model a list of associated engineering data. A query is then made of an external source to retrieve values for the engineering data for the time interval selected by the user.

The final part of the problem description provided by the user is the rule base to be applied to reason about the device model. Each model knows what rule bases are associated with it, and the list of associated rule bases is provided to the Interface Management System for presentation to the user via a menu.

Three major design criteria implemented by the Interface Management System are to:

- provide a point-and-click style of interface that minimizes use of the keyboard and maximizes use of the mouse,
- provide a set of reusable display objects that give a consistent look and feel across applications, and
- establish a protocol that application developers may follow to access and use the display objects.

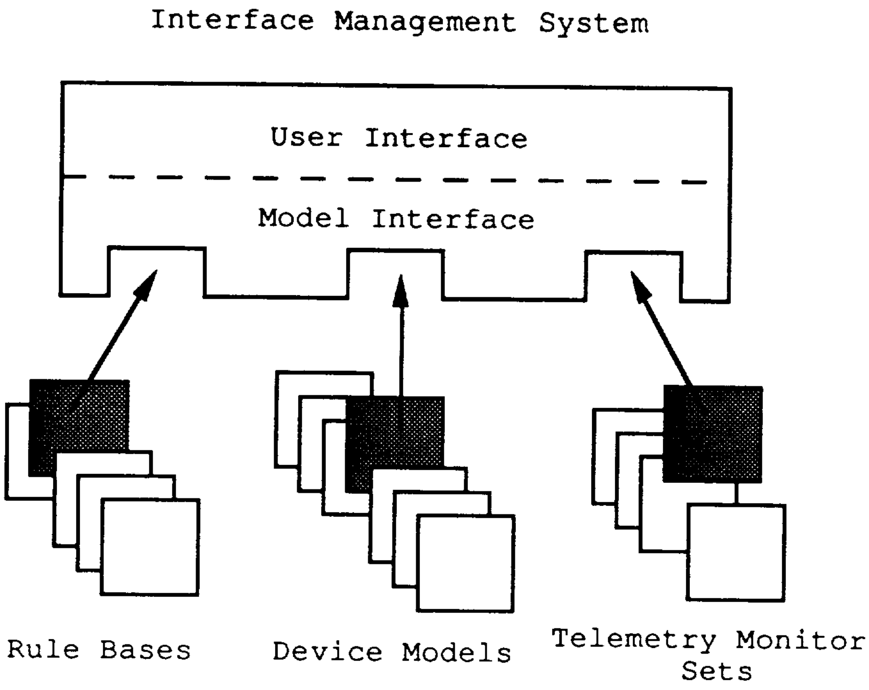

Figure 2. The HSTORESIS Concept

Examples of the types of display objects that are available include buttons, query panels, menus, telemetry monitor display panels, message windows, and time displays. Graphical images are also included. For example, one graphical image depicts the orientation and location of the HST relative to the earth, moon, and sun. More will be said later about the protocol available to application developers for using display objects.

HSTORESIS implements a display page library. A display page consists of a collection of display objects accessible by the user. Display pages may be built by a user and saved in a display page library. Display pages are indexed by user name and problem type. A user may build, modify, and save any display pages owned by the user. Display pages owned by other users may not be changed, although pages belonging to other users may be copied into the current user's work space and then modified, if desired.

The complete set of display objects provide a powerful tool for use by the application developer. The objects are implemented using object oriented programming techniques and can be accessed via a simple messaging scheme.

The Interface Management System manages the creation and display of the objects, the collection of answers from the user, and the return of the user inputs to the messaging object. Two important benefits derive from this. First, the amount of interface programming that an application developer must do is significantly reduced. Second, the reuse of the set of display objects provides a consistent look-and-feel for the user across problem solving sessions.

Reusable Telemetry Database

The source of information about the behavior of the HST is engineering data obtained from monitors on board the HST. Data is collected and communicated via telemetry to a ground station. There are approximately 5,500 telemetry monitors associated with the HST. The interesting technical issues concerning the monitors are how to represent them in a knowledge base and how to obtain descriptions of the monitors associated with a device model.

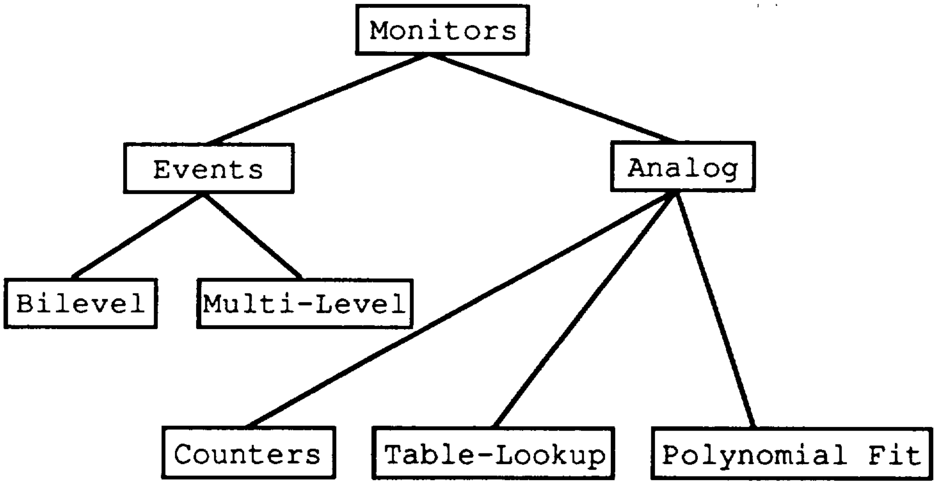

Figure 3. Monitor Classes

The solution to the representation issue is illustrated by the design in Figure 3. All 5,500 telemetry monitors conform with the design.

Event monitors have values that are either bi-level or multi-level. Bi-level monitors have measurements with only two states (e.g., on or off). Multi-level monitors have measurements with more than two states (e.g., high, medium, or low).

Analog monitors are either counters, table look-ups, or polynomials. Counters represent uninterpreted telemetry counts. If a counter has a value of three, then three is the correct meaning of the value. A table look-up requires retrieval from a table of the analog value corresponding to the monitor value. For example, a monitor value of three might correspond to an analog value of 1.67 volts. A polynomial fit is a monitor whose value is inserted into a polynomial equation. The meaning of the monitor's value is the solution of the polynomial equation. For example, if a polynomial equation for a monitor has the coefficients of 1.5, 0.03, and 2.1, then a monitor value of 3 has the meaning:

$$1.5*3^0 + 0.03*3^1 + 2.1*3^2 = 20.49.$$

Given the design, the problem of extracting the descriptions of the desired telemetry monitors becomes simply a matter of generating a list of the desired monitors, locating them in a master database of monitor descriptions, and then creating a knowledge base from the descriptions of the monitors. All of this is automated by HSTORESIS which eliminates the need for manually producing the descriptions.

Reusable Device Models

Within HSTORESIS, a satellite telemetry point is represented as an object with its own data and set of behaviors. In one sense, the instantaneous state of the HST is represented by the collective output of its 5,500 telemetry monitors. However, this representation is extremely weak since it lacks information about component connectivity and component behavior. Although rules can be written that reason exclusively in terms of telemetry values, human experts do not usually think in these terms.

By extending the above analogy one step further, the state of each major component of the HST is represented by the values of some set of monitors. The mapping between a set of monitors and a component forms the nucleus of device models used in HSTORESIS.

The monitor mappings, however, are only part of the model abstraction. A complete device model will include all of the following:

- a mapping between the model and a set of monitors,
- pointers to the rule bases that are capable of reasoning about the device,
- behaviors (methods) that represent the conceptual or physical functioning of the device/component, and
- features (slots) that hold state information that is not included in the satellite telemetry stream.

The first two items are used dynamically to establish bindings between the HST telemetry database, the HSTORESIS device model, and the user interface. The final two items encode the procedural knowledge that relates to the device. For example, if a reaction wheel were being modelled, an example of the final two items might be a method for computing angular momentum, or features like the wheel's mass or composition.

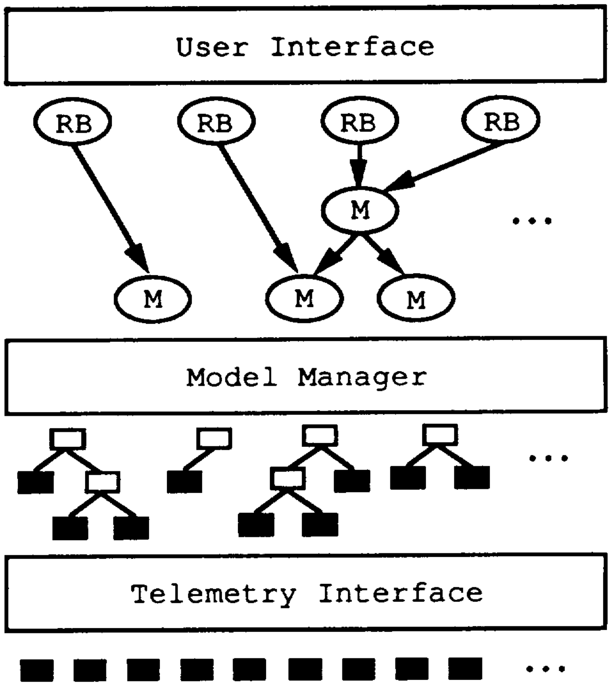

Figure 4. HSTORESIS Architecture

Figure 4 illustrates how these ideas have been incorporated into HSTORESIS to transform raw telemetry data into a level of abstraction which is closer to the mental representations that human experts use. The telemetry interface stands between the raw telemetry data and the telemetry object abstraction. The model manager lies between the telemetry objects and device models (indicated by bubbles labelled with an M). The user interface layer provides the user with direct access to the rule systems that reason over device models.

Graphical Object and Rule Integration

An important feature of the Interface Management System, mentioned previously, is the protocol for accessing graphical objects. The protocol supports development of rules that can both deliver information to and solicit information from the user. The use of graphical objects can be described by referring to Figure 5, which depicts a schematic of the screen layout for HSTORESIS.

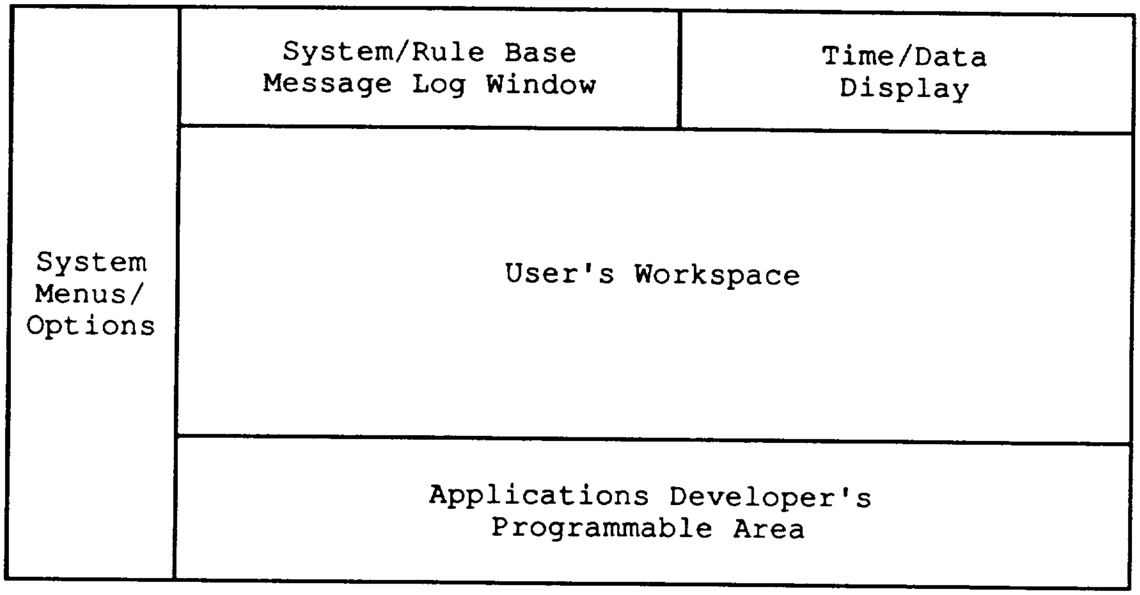
System/Rule Base Message Log Window
Time/Data Display
System Menus/ Options
User's Workspace
Applications Developer's Programmable Area

Figure 5. HSTORESIS Screen Layout

The System Menus/Options area permits the user to access application independent menus and options for defining a problem set, editing the workspace, and launching and managing a query. Queries that can be made by a user are determined by the rule base selected by the user to be applied to a set of telemetry data. User queries are controlled by start, pause, and step options. The basic visual metaphor for interacting with this area is a button.

The Application Developer's Programmable Area is reserved for the display of application specific options. The application developer is provided with a simple protocol for programmatically displaying and removing options that are application specific. Again, the basic visual metaphor for user interaction with this area is a button. An application developer may choose to provide buttons in this area that correspond to a set of queries that a rule base can answer about the telemetry data to be analyzed.

The System/Rule Base Message Log Window area is reserved for the display of system messages (e.g., warnings) and the results of inferencing. The Log Window provides the user with icons for controlling which message is displayed. Messages may be scrolled through the window, or a specific message may be selected for display. A counter indicates how many messages are available and the number of the message which is currently displayed.

The Time/Data Display depicts three items. The spacecraft time is indicated. The frame of the telemetry data that is currently being analyzed is indicated. A slide bar indicating the percentage of telemetry data that has been (or remains to be) analyzed is also provided.

The User's Workspace allows the user access to display objects that can be used to monitor the state of the device model as the analysis session progresses. Some of the objects that may appear in this area are monitor display objects, graphical images such as the one used to depict the relationship of the HST to the sun, earth, and moon, and pop-up objects such as query panels. Most of the interaction between the user and HSTORESIS will occur in this area.

Pop up query panels permit rules or other objects to ask questions of the user. Query panels are also used to obtain information from the user at the start of the analysis session. The user must provide a user name, a device model, and a time interval over which the analysis is to be made.

Pop up dialog boxes permit rules to provide information to a user. Optional buttons may be associated with a dialog box to provide additional capabilities for the user. Figure 6, for example, depicts a rule that creates a pop up dialog box with a button labelled "Recovery" which permits access to information about recovery from an event called "RWA-Speed-Limit-Test".

```
(IF
  (TEXT (RWA-SPEED-LIMIT-TEST HAS FIRED))
  THEN
  (LISP
    (UNITMSG '(SIS-SCREEN-MANAGER SIS-SCREEN-MANAGER-KB) 'IN-BOX
             'DISPLAY-POP-UP-MESSAGE :TEXT
             "RWA Speed Limit Failure: check for:
             ~%Too large vehicle maneuver
             ~%momentum management performance
             ~%misconfigured software
             ~%other database problems."
             :BUTTON-VALUES
             '(("Recovery"
                '(UNITMSG '(SCREEN-MANAGER SCREEN-MANAGER-KB) 'IN-BOX
                          'DISPLAY-POP-UP-MESSAGE :TEXT
                          "1. Monitor wheel speed in sun point until
                              it returns to normal.
                         ~%2. Dump memory to re-verify the database
                         ~%3. Work through Section 8.0 recovery
                              procedure."))))))
```

Figure 6. Sample Rule

A more interesting use of pop up dialog boxes is to provide a nonlinear text or hypertext functionality. For example, the rule in Figure 6 could be modified to provide a button connected to a function that could access the "Section 8.0 recovery procedure" in the appropriate design document. Then, reviewing the recovery procedure would be as simple as clicking on a button. An obvious extension to this capability is providing access to a video disc containing schematic drawings or other graphical images pertinent to the analysis being performed.

Conclusions

HSTORESIS demonstrates a successful approach to integrating knowledge from multiple domain experts into a single knowledge base system. An adaptive, knowledge-based interface facilitates interaction between a user and domain specific rule and knowledge bases. The application demonstrated by HSTORESIS is analysis of safemode events, which is diagnosis problem. However, HSTORESIS could easily be extended to other applications such as training, scheduling, design, etc. Additionally, HSTORESIS provides a capability for accessing on-line design documents in a nonlinear manner. This allows the user to access design knowledge not specifically contained in the HSTORESIS knowledge bases.

Acknowledgement

The work described in this paper has been performed under the NASA Hubble Space Telescope Contract number NAS8-32697, Modification 695.

References

1 "Systems Autonomy Technology Program (SATP) Plan," (FY89 draft), NASA/Ames Research Center, p. A-160, August 1988.

2 Freeman, M. S. "The Elements of Design Knowledge Capture." Proceedings of the Fourth Conference on Artificial Intelligence for Space Applications, pp. 39-46, November 1988.

3 Genesereth, M. R. "The use of design descriptions in automated diagnosis." Artificial Intelligence, 24:411-436, 1984.