The Space Congress® Proceedings

1984 (21st) New Opportunities In Space

Apr 1st, 8:00 AM

LOX Expert System

Ethan A. Scarl
Knowledge Based Systems, The MITRE Corporation, Bedford, Massachusetts

Carl I. Delaune
Software Applications Branch, John F. Kennedy Space Center, NASA

LOX EXPERT SYSTEM

Dr. Ethan A. Scarl
Knowledge Based Systems
The MITRE Corporation
Bedford, Massachusetts

Dr. Carl I. Delaune
Software Applications Branch
John F. Kennedy Space Center, NASA

ABSTRACT

The LOX Expert System is a computer program which uses artificial intelligence (AI) techniques to diagnose instrumentation problems in the shuttle liquid oxygen fueling system. The KNOBS knowledge-based system is being modified for application to this problem. System functionality and fault isolation methods are described.

INTRODUCTION

Recent studies (Saga80, Benz82) have concluded that NASA could perform more productive missions at lower cost by making better use of computer science. One discipline of computer science emphasized in these reports and in the NASA Computer Science Research Program Plan is that of AI. An investigation (Akin83) of potential uses of AI at the Kennedy Space Center (KSC) has identified Expert Systems as a powerful, maturing technology which can be profitably applied to assist in launch operations.

Space Shuttle launch operations require knowledge of special purpose (often unique) equipment and procedures. This body of knowledge, acquired over decades, is a valuable resource of the shuttle program. Expert Systems technology offers the capability of retaining this knowledge, in usable form, in computer programs. In addition, Expert System programs may perform some critical tasks faster and more reliably than humans, making them candidates for several applications in launch operations.

The LOX Expert System (LES) is being developed as a tool for constraint-based monitoring and analysis of propellant loading. This is the first application of AI techniques at KSC. LES will be used to evaluate the utility of Expert Systems for this and other NASA applications.

LOADING LOX AT KSC

The Shuttle Orbiter is powered by three main engines which burn liquid hydrogen and oxygen (LOX). These propellants are carried in a large disposable external tank (ET) which is jettisoned after main engine burn, whereafter these engines are dead weight to be carried back to earth.

Loading the ET is a major prelaunch operation. Volatile cryogenics cannot be loaded long before launch and must be continuously replenished until seconds before liftoff. The normal loading operation commences about six-and-a-half hours prelaunch and goes into replenish mode three hours later. 140,000 gallons of LOX and 390,000 gallons of liquid hydrogen fill the ET from storage tanks over a third of a mile away from (and well below) the launch pad, all some two miles from controlling computers and operators at the Launch Control Center (LCC). Of the two, LOX is the more complex and troublesome operation (in spite of the hydrogen's far colder temperature), principally because the liquid hydrogen is so light that it can be transported by pressure from evaporators faster than the LOX is mechanically pumped. Pumping large volumes of cryogenic liquid over long distances is complex and hazardous. With no more than two hours slack time to recover, a serious loading problem could lose the launch window and cause the entire flight to be replanned.

THE LAUNCH PROCESSING SYSTEM

Fuel loading operations are controlled and monitored from the LCC by the Launch Processing System (LPS), a real time process controller running on a network of minicomputers. LPS application programs monitor pressures, temperatures, and flow rates, and control valves, pumps, and other hardware. The LPS was designed with redundance and conservatism. Although the intended redundance was to allow the launch to

continue should any single component fail, conservatism dictates that when certain critical measurements deviate from their expected ranges, the entire system is "safed." This means that the current operation is brought to a stable configuration and stopped there until the reason for the deviation is determined. The detection of anomalous measurements and subsequent system safing are performed automatically by the LPS' reactive control logic (RCL), but the troubleshooting is manual.

FOCUSSING THE PROBLEM

Historically, shuttle launches have been threatened more by instrumentation problems than by actual system failures, and there is at present no fast, reliable way to distinguish between these. When a non-trivial problem occurs, system engineers pore manually through schematic diagrams and, given time, will isolate the failure to hardware, software, or instrumentation.

LPS' RCL may commit to a safing procedure in seconds, but it may take a long time to recover from that decision. One would like to recognize the sensor failures in time to prevent safing, but such fast diagnosis requires mechanical help. Given the complexity and size of the system and the reaction time required, the cost of error is unacceptably high.

Furthermore, the current procedure relies on the expertise of a few key individuals, whose accumulated experience is nearly irreplaceable. Contract changes and normal attrition processes will gradually thin the ranks of these talented, valuable people. It would be most desirable to capture this expertise before it is lost.

These considerations have focussed the current effort on sensor problems. Once sensor failures are subtracted out, the existing LPS software is considered adequate to determine when the physical loading procedure has gotten into trouble.

KNOBS

This section will briefly survey some of the components of the KNOBS system. More complete descriptions of KNOBS and its other applications can be found elsewhere (Enge79, Enge80, EnSt81, PaEn83, Enge83, Mogi83).

A set of labelled items to be provided with values is the simplest case of a structure called a frame. Frames (Mins75) generalize the concept of a property list for an object, or a set of attribute-value pairs, as a natural format for storing knowledge about the object. KNOBS uses frames both for general factual knowledge (e.g., valves, measurements, etc.), and to represent its current plans and situation descriptions. The Frame Representation Language (FRL), one of several implementations of frames, was developed at MIT by Roberts and Goldstein (RoGo77). Implementing a particular model of frames with functions to create and manipulate them, FRL has been considerably extended and modified KNOBS. We will generally use frames to represent objects and their attributes (called "slots"). The values of slots may be other frames, thus forming a linked network in which slots represent the relations between objects.

KNOBS also uses "rules," small chunks of knowledge coded in if-then format, but less so than most popular expert systems architectures.

In the planning type of problem to which KNOBS has been heretofore applied, the user interacts with the system by making choices which are then criticized by constraints (figure 1). These constraints are automatically triggered when the plan slots that they constrain are filled or changed. Demons are triggered in the same manner, but perform useful actions such as updating the data base or filling other slots. The system can be asked to help by the listing choices for a given slot which are compatible with the current situation, and perhaps applying preferential criteria to order them. Or it may be able to offer a consistent set of selections found by a dependency-directed backtracking algorithm (in effect, automatic plan completion). Inference rules may be invoked by the constraint checking, or the choice ordering, and automatically generate explanations of their conclusions. Some slots, marked "automatic," have values that can be calculated in a routine, automatable fashion and may be filled by the user or automatically by the system (using demons) as soon as the slots on which they depend are given values. In the original Air Force application, English can be used not only to answer questions in the current context by retrieving information from the database, but to add to the database, modify rules, or govern the entire planning process.

Not all these facilities have been developed for every application, and the application described here is sufficiently different that their transfer may involve considerable innovation.

THE DATABASE

The system must be able to represent and access information now contained in schematic diagrams. The portion of the LPS relevant to liquid oxygen loading alone includes thousands of replaceable components, described by more than 200 schematic diagrams. At the core of LES is a database of FRL frames. Each major component in the instrumentation system is represented as a frame. A frame contains information about a particular

component, general information true of a whole class of components, measurement values associated with LPS-interfaced components, and the relationship of the component to its power sources and to the other components that control it or are controlled by it. Thus, an automatic traversal through a chain of frames shows how a component is ultimately controlled, which might previously have been only discovered by leafing back and forth through several pages of schematic diagrams. These same frames also contain information showing nominal values of measurements and the expected relationships between values expected for different measurements.

The input to LES is a sequence of time-tagged measurements from the LPS. In its initial phase, historical data from actual Shuttle launches are being supplied from magnetic tape. In a future operational mode, data could come in near real time from a Central Data Subsystem (CDS) which performs archiving functions for the LPS.

THE ROLE OF THE KNOWLEDGE-BASED SYSTEM

The system monitors data through the LPS from the liquid oxygen loading operation and checks it for consistency. Time and source-stamped readings indicate temperatures, pressures, flow rates, liquid levels, voltages, currents, and valve positions. LES does NOT attempt to determine whether the operation is proceeding correctly (the logic for doing that is well-documented and already embodied in LPS code), and does not even know the official state of the loading process (chilldown, fast fill, etc.). LES only tries to determine whether or not the LPS is being told the truth by its sensors. Incoming data is tested by constraints which compare it against expectations. These expectations are generated by a system model implicit in the frames and the other related measurements. When an anomaly is detected, the LPS and monitoring personnel are notified, and troubleshooting algorithms are activated.

The choice of problem being attacked here has the attractive characteristic that it is building on top of an established system which performs acceptably well in normal circumstances, so that the new system can be phased in incrementally and tested in the working environment with minimal risk to current operations. Eventually, LES could be hardwired to signal a sensor failure directly to the LPS.

If a definite culprit is isolated, LES will have fulfilled its duty. If a culprit cannot be found (and LES should be able to find one as often as a human engineer), then it offers a list of suspect components. Rather than assigning probabilities among these alternatives, explicit instructions will be offered for performing tests which will result in isolating the culprit (e.g., looking at an indicator not interfaced to the LPS, training a TV camera onto a gauge out on the mobile launching platform, or sending a crew out to test a junction point with a voltmeter.

The relationship which has existed between KNOBS and its users in all previous applications is here dramatically altered (figure 2), because the key operation is one of monitoring rather than planning. The LPS supplants the user as the source of new values to be tested. There is no longer a role for suggesting choices or planning automatically, since the physical world is already making those choices without listening to KNOBS' advice. But we still need KNOBS' constraints to criticize choices, and demons to respond to them. We also need the explanation facilities for these criticisms, and must extend such explanations to cover diagnostic responses.

Two operating assumptions are used throughout:

a. The Single Point Failure Hypothesis. Any new problems is assumed due to a single failure. Any interpretations requiring two or more new and simultaneous failures will be discarded. Naturally, once a fault is isolated, it will be so marked (eventually, LES will send LPS a command to "bypass" the defective component; for now an operator must intervene manually to do this) and thus not confound the analysis of later failures.

b. The Steady State Hypothesis. LPS' RCL currently operates only when the system is in a well-defined flow state, and LES will do likewise. This not only sidesteps problems of recognizing and interpreting transient physical conditions, but also temporary inconsistencies due to one measurement being sampled a full second before another.

REPRESENTING COMPONENTS THROUGH "SOURCE-PATH-SINK" ANALYSIS

Each major LOX system component is seen as transmitting control. One component may be controlled by certain others, and controls still others in turn. At each step, the control signal is translated from one medium to another. A relay may translate low current to high current, or an off condition into an on condition. A pressure transducer translate a fluid pressure into a voltage. Some valves translate voltage into gas pressure, while others translate nitrogen gas pressure into liquid oxygen flow. In general, each such control step is supplied some medium (e.g., a power supply current, or maybe a flow of gaseous nitrogen, which we call a "source". A component is implicitly

a "sink" for its source.

The source energy or medium is modulated by a control signal to determine the state of the component. We refer to such controlling components as its "SOURCE-PATH." In more complex cases, the SOURCE-PATH holds an expression which not only lists the components exercising control, but which makes explicit the way they combine to exert that control (e.g., which switches must be in their power-off positions, and which must not be, in order for this component to deviate from its own power-off position).

This perspective allows us to represent the functionality of the LOX control circuitry to the required level of detail, without having to encode its full schematic diagram and then having to extract control information from it when the need arises.

Figure 3 indicates the frame database representation of a component call ed GLOP2026A, which is a measurement of pressure being sent back to the LPS. GLOP2026A happens to be a critical pressure reading at the orbiter inlet of the mobile launching platform, and the RCL will safe the system if GLOP2026A goes out of bounds. Its source-path is marked "TRUE" to indicate that this component requires no external power source (and hence that source can never fail). GLOP2026A accepts a signal in DC VOLTS from the pressure transducer marked A86470 (whose power source is +D180A), and translates it into a PSI reading. Not shown is the part of the LPS to which this psi reading is transmitted.

This scheme is handled a bit differently with system components which are less concerned with control than with the distribution of some source medium, like DC power or gaseous nitrogen pressure. For example, +D180A is interpreted as representing a fuse, and its source in turn is some circuit breaker. +D180A has an explicit "SINKS" slot which lists all components depending upon it for power.

The most current measured value of GLOP2026A is the value in its CVALUE slot, and the time of the measurement is also recorded.

EXPECTED VALUES

KNOBS has traditionally handled constraint triggering by maintaining "constraint references" in the CONSTRAINTS slot of a "template" frame (PMT in figure 3) retrieved through the individual frame's generic heirarchy (the PRESSURE-MEASUREMENT frame of figure 3). In LES, constraints are ultimately statements of the consistency of a measurement with all other such measurements, although they may be expressed in terms of local objects which are not directly measurable (but whose statuses are in turn traceable to one or more measurable items). Thus the expected STATUS of GLOP2026A may actually be derived from that of A86470, which is controlled by the LOX pressure in the LOX transfer line, which in turn can be thought of as deriving from an upstream pipe pressure whose measurement is called GLOP2016A. This is all condensed into figure 3 by showing the STATUS OF GLOP2026A as a simple function of that of GLOP2016A (subtracting 20 psi), to within a tolerance of 10 percent.

Discrete (on/off or open/closed) measurements naturally have no tolerances. Often they may need no STATUS slot either, since the SOURCE-PATH information will determine an expected value.

Here we have another radical difference from KNOBS' planning applications, most of which required substantial numbers of constraints referenced by a relatively small number of templates which control the display and testing of the different partial plans. In this system, the template PMT is actually attached to a much higher level frame (it is really a Physical-Measurement Template). There is only one constraint which tests an individual component's CVALUE against the value expected from its STATUS (or SOURCE-PATH), and one demon which runs the diagnoser (below) when the constraint fails. This reflects a substantial shift in the representation of domain knowledge from constraints to frames.

Certain types of sensor are plentiful (such as temperature and pressure) along the fluid flow path, and can be tested against adjacent sensors of the same type. For example, with steady state flow one expects all pressure readings to drop monotonically along the LOX pipe from the pump to the ET (due more to increasing elevation than to viscous flow). Similarly, the liquid temperature is expected to rise monotonically from the storage tank to the ET (although the ET ullage pressure may be lower, due to evaporation). These monotonicities may be all the check one needs, and a suitable constraint on GLOP2026A may be merely that it is less than GLOP2016A.

If more accurate tests are desired, they must be constructed from either physical or historical knowledge about the system. For systems such as the LOX loading system where there exists a substantial reservoir of experience, it is probably better to encode experiential knowledge of what a reading should be in terms of its neighboring measurements than to calculate it from first principles. In the future, LES may keep and use its own historical information to upgrade its expectations, as a simple learning technique.

THE DIAGNOSIS ALGORITHM

Once some sensor has failed, a limited deduction is initiated to determine the cause of failure as specifically as possible, using knowledge of the LOX system's power and control relationships. Bear in mind that these constraints are designed not to fail unless the problem can be definitely attributed to a sensor and not to an unexpected physical measurement. So, while it may remain to be determined which one, a decision has already been reached that the problem is with some sensor defect.

Each object is initially classed as "innocent," but may be reclassified as a "suspect" or the "culprit." By the single-point failure hypothesis, any defective component is the only culprit, and must be able to explain all failed constraints. (We are presently ignoring any possibility that one component's defect may be "masked" until some other component fails.) If we end up with a list of suspects instead of a single culprit, then each of those suspects must by itself potentially explain all constraint failures.

Some measurements are completely redundant in that two or three output devices (which we call "clones") are connected together directly and should read identically at all times. Suppose there are three or more clones: if none disagree, then all are innocent but must have their source and source-path checked. If one disagrees with the others, then it is the culprit. No more than one can be wrong, by the single-point failure hypothesis. If there are only two clones, and one disagrees with its STATUS, then that is the culprit. If there are no clones, then the object is suspect if it disagrees with its STATUS expectation, and innocent otherwise.

If the component disagrees with its status expectation, then the power-off status of the component is retrieved (usually inherited from its generic type). If the current value (CVALUE) is equal to the power-off value (to within TOLERANCE), then the component's SOURCE is tested. This means testing all the SINKS of this source (except the original object); if a sink is found with any power-on value, then this source is innocent.

If a sink is found to have its power-off value in violation of its expected value, then we know that either this source or the source of this source is the culprit, and the source's own source must be checked similarly. If the source's source is good, then the source itself is the culprit.

If no sink of the source is on or wrongly off, then the source remains suspect, and if no other culprit is determined then the diagnoser's output will include instructions to try turning on one of its sinks or to test it directly with a meter.

Sources are treated just like any other objects by the diagnoser, but they respond differently since only they have SINKS.

It remains to test the object's SOURCE-PATH. All objects mentioned therein are treated as suspects. Many objects (like A86470 in figure 3) are not interfaced to the LPS and therefore have no CVALUEs. Such objects cannot be checked directly, but become suspects and have their SOURCE-PATHs checked.

If no culprit is found after checking all components implicated by constraint failures, then suspicion becomes focused on those objects which are labeled as suspects by all failed constraints. If there are none, then there is an error condition for the diagnoser. If there is one such object that is the culprit. If there are more than one then diagnosis instructions are printed out for Shuttle personnel. Presently, there are separate instructions stored with each object type, and people are told to test them one by one until the culprit is located. Later, we hope to merge these instructions more intelligently into a decision tree.

ACKNOWLEDGEMENT

Neither LES nor the KNOBS system itself could possibly exist without the accumulated insight, energy, and perseverance of Carl Engleman. This work is one of several ongoing efforts which serve as living monuments to his memory.

(Akin83) D.L. Akin, M.L. Minsky, E.D. Thiel, "Automation and Machine Intelligence Applications to Selected Launch Operations," Unpublished.

(Benz82) W.L. Benzon et al, "Computer Science: Key to the Space Program Renaissance," Final Report of the 1981 NASA/ASEE Faculty Summer Study Group, June 1982.

(Enge79) C. Engelman, C.H. Berg, and M. Bischoff, "KNOBS: An Experimental Knowledge Based Tactical Air Mission Planning System and a Rule Based Aircraft Identification Simulation Facility," Proc. Sixth Inter. Joint Conf. Artificial Intelligence, Tokyo, 1979, pp. 247-249.

(Enge80) C. Engelman, E.A. Scarl, and C.H. Berg, "Interactive Frame Instantiation," Proc. First Annual Nat. Conf. Artificial Intelligence, Stanford U., August, 1980, pp. 184-186.

(Enge83) C. Engelman, J.K. Millen, and E.A. Scarl, "KNOBS: An Integrated AI Interactive Planning Architecture," in Proceedings of the American Institute of Aeronautics and Astronautics, Computers in Aerospace IV, Hartford, 1983.

(EnSt81) C. Engelman and W.M. Stanton, "An Integrated Frame/Rule Architecture," in Proceedings of NATO Symposium in Human and Artificial Intelligence, Lyon, October, 1981.

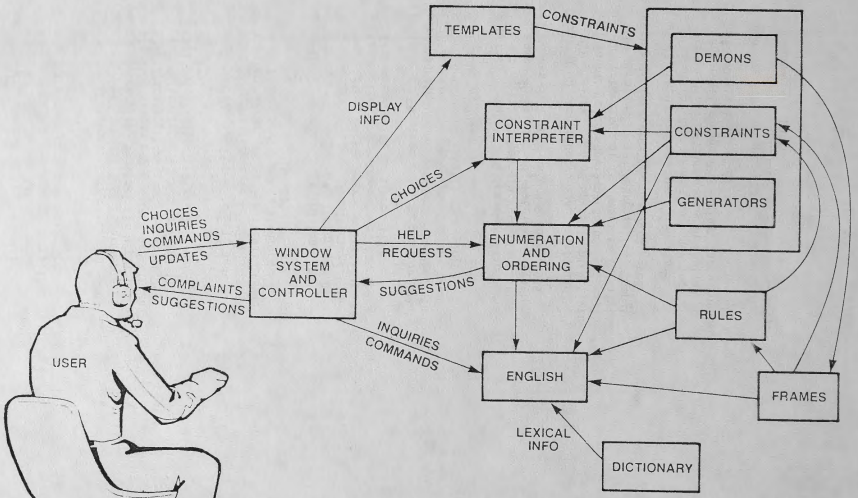
(Mins75) M. Minsky, "A Framework for Representing Knowledge," in Psychology of Computer Vision, ed. Patrick H. Winston, McGraw-Hill Book Co., New York, 1975.

(Mogi83) J. Mogilensky, E.A. Scarl, and R.E. Dalton, "Manned Spaceflight Activity Planning with Knowledge-Based Systems," in Proceedings of the American Institute of Aeronautics and Astronautics, Computers in Aerospace IV, Hartford, 1983.

(PaEn83) M.J. Pazzani and C. Engelman, "Knowledge-Based Question Answering," Proceedings of Conference on Applied Natural Language Processing, Santa Monica, February, 1983.

(RoGo77) R.B. Roberts and I.P. Goldstein, "The FRL Manual," MIT AI Lab., Memo 409, September, 1977.

(Saga80) Carl Sagan, Raj Reddy, Edward Feigenbaum et al, "Machine Intelligence and Robotics," Report of the NASA Study Group, March 1980.


TEMPLATES
CONSTRAINTS
DEMONS
DISPLAY INFO
CONSTRAINT INTERPRETER
CONSTRAINTS
CHOICES
GENERATORS
CHOICES
INQUIRIES
COMMANDS
HELP
REQUESTS
ENUMERATION AND ORDERING
UPDATES
WINDOW SYSTEM AND CONTROLLER
COMPLAINTS
SUGGESTIONS
SUGGESTIONS
RULES
INQUIRIES
COMMANDS
USER
ENGLISH
FRAMES
LEXICAL INFO
DICTIONARY

2-22

Figure 1

Launch Processing System (LPS) Integrated LES Configuration

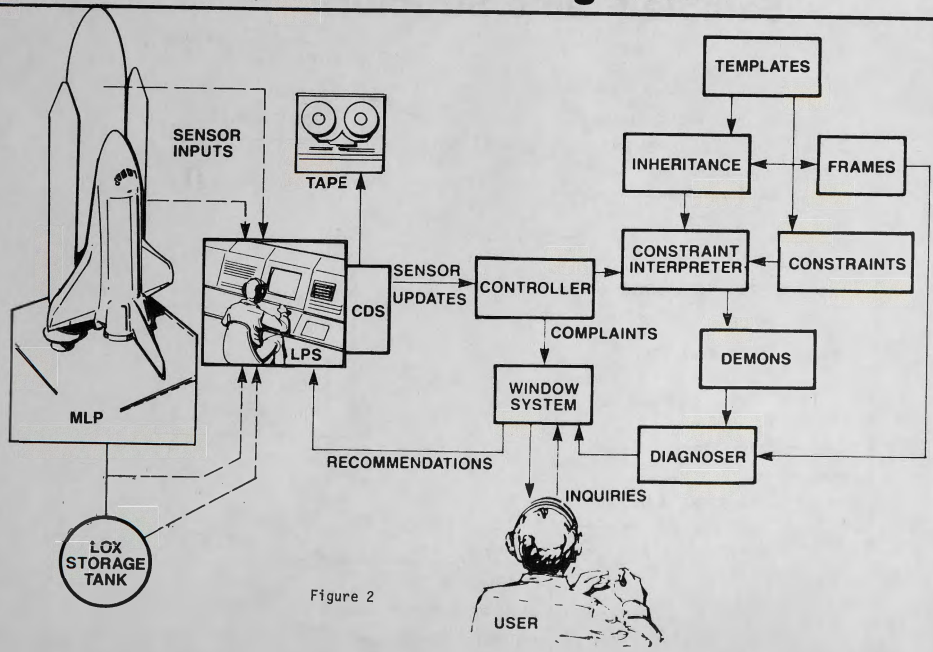


Figure 2

Sample Frame Structure

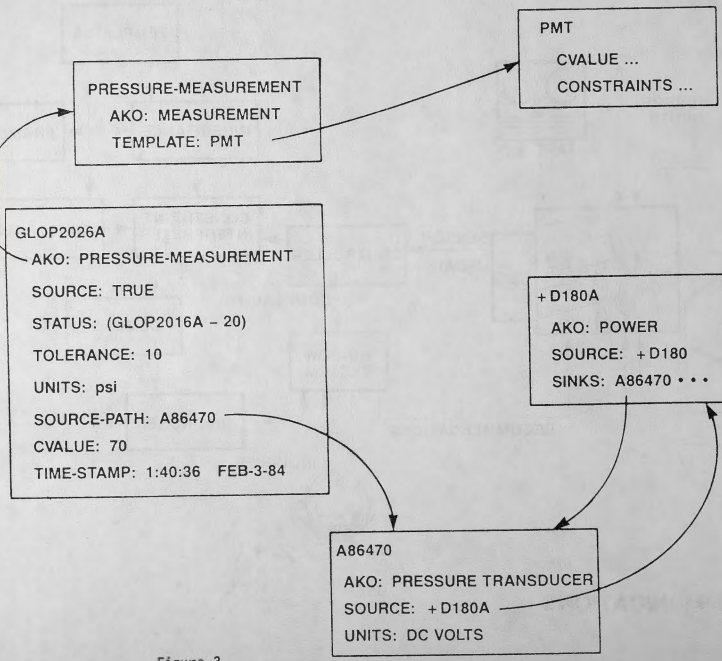


Figure 3