EXPERT SYSTEMS IN DATA ACQUISITION

Bob McCauley
Computer Sciences Corporation
Telemetry Systems Operation
Lompoc, CA

ABSTRACT

In an Independent Research and Development (IR&D) effort, the Telemetry Systems Operation (TSO) of Computer Sciences Corporation (CSC) sought to determine the feasibility of using Artificial Intelligence (AI) techniques in a real-time processing environment. Specifically, the use of an expert system to assist in telemetry data acquisition processing was studied. A prototype expert system was implemented with the purpose of monitoring F15 Vertical Short Take Off and Landing (VSTOL) aircraft engine tests in order to predict engine stalls. This prototype expert system was implemented on a Symbolics 3670 symbolic processor using Inference Corporation’s Artificial Reasoning Tool (ART) expert system compiler/generator. The Symbolics computer was connected to a Gould/SEL 32/6750 real-time processor using a Flavors, Inc. Bus Link for real-time data transfer.

KEY WORDS

Artificial Intelligence
Expert Systems
Aircraft Engine Testing
Expert System Compiler/Generator
Inference Engine

INTRODUCTION

The objective of our expert system was to computerize the thought processes of Flight Test Engineer experts. The expertise could then be applied to situations requiring the objectivity of a computer. This technique also allows valuable expertise and knowledge gained throughout a professional’s career to be captured in a computer program and used by others who must perform similar or complementary job functions. Expert systems can contain and use large stores of facts, concepts, rules-of-thumb, and other types of expert knowledge. The sought after effect is to reduce training and operating costs. Due to the complexity of the human brain, however, the building of these expert systems is not always straightforward. Additionally, due to the relative late emergence of this technology, few areas of activity have been investigated and documented as to the applicability of expert systems.

This study was proposed, not to develop a working system to be deployed, but rather to investigate the feasibility of this technology. The specific area of monitoring the health and welfare of the F-15 VSTOL engine tests was chosen because of the access to experts in this field.

This paper is divided into two areas of concern. First, a look at the representative generic architecture is made. Second, there is a discussion of implementation problems, as well as the difficulties associated with extracting the “expert information” from the experts.

DEVELOPMENT SUITE

The expert system development suite is shown in Figure 1 and includes the following components:

a. Symbolics 3670 Computer. This symbolic processor provided the basis of development of the expert system activities. This processor was chosen because of its widespread use and acceptance in the industry. This system came complete with a wealth of development tools required to facilitate the development of rule-based programs.

b. Artificial Reasoning Tool (ART). This expert system generator provided an easy to learn language that permitted rapid prototyping of the rule-based processing, along with easy to use capabilities for the generation of real-time displays. This language was chosen because of its wide-spread use and acceptance on similar projects. This language provides the user with forward chaining, backward chaining and viewpoint analysis capabilities.

c. LISP. This language was used to provide low level access to operating system interfaces. The use of this language provided more rapid data transfers for input/output operations. An appropriate analogy is that LISP is to ART as assembly language is to FORTRAN.

As seen in Figure 1, the domain expert, who is the person that possesses the expertise required by the application, works with a knowledge engineer to define expert rules and procedures using ART and LISP. Preliminary testing of the generated expert system was done on the Symbolics computer using the debugging capabilities provided. This debugging function allowed the tracing and monitoring of logic steps using convenient, easy-to-use display windows.

Complex activities, such as monitoring high speed engine tests, involve the resolution of conflicting determinations. The expert system, in the same fashion, must have the ability to compare these conflicts, and provide the highest priority to one set involved in the conflict. This effort of resolving conflicts is an inherent capability of the ART inference engine. The identification of problems is closely related to the forward chaining capability of expert systems such as are produced by the ART system. The backward chaining of ART provides the capability of working backward from a problem to determine the cause of the situation. The expert observer adds flight test expertise from having observed many such tests. ART also provides the capability to add new rules to the system as a result of rule processing. This, in effect, allows the system to automatically "get smarter". The final effect of the Expert System when reacting to a verified anomaly is to provide a corrective action. In order to provide a higher degree of trust in such a computer-generated action, the ART viewpoint analysis capability can be used. This feature provides a "what if" type walk through of possible ramifications of the implementation of the proposed corrective action.

It should be stated that most of the described expert system generation capabilities of a system like ART are available in traditional arithmetic processors. However, the loose coupling of rule set architecture in logic processors permits the addition of rules as the system matures without excessive restructuring of the programs, such as is the case with traditional non-expert systems.

VALIDATION ENVIRONMENT

After the expert system prototype was developed and checked out, more thorough validation was possible by passing data from the Gould/SEL real-time data acquisition system to the Symbolics computer over a Flavors Bus Link interface. The Bus Link provides this transfer capability by allowing the physical memory space of the Symbolics processor to be mapped into the physical memory of the Gould/Sel processor. In this manner, the mapped memory is used as a shared memory (accessed by memory reads/writes), such that, after initial semaphore communication is established, explicit input/output instructions are no longer necessary. This is similar to techniques used by array processors such as the Sky Warrior. An effective data transfer of 4 megabytes per second is provided by the Bus Link. Random access to all mapped memory addresses is provided.

Figure 2 depicts the data path of information from a data tape to the Gould processor, after being processed by the real-time data acquisition system, data is then passed over the Flavors Bus Link interface to the expert system in the Symbolics processor.

EXPERT SYSTEM ARCHITECTURE

Figure 3 illustrates the expert system architecture that was used to implement the prototype. A LISP-based input/output task fields the data that is transferred over the Bus Link interface to the Symbolics processor. The use of LISP provided optimization at a potential choke point. The architecture of ART provides the flexibility of driving the expert system with a LISP module, or having the expert system call the LISP module in subroutine fashion. For this study, the LISP module functioned as the master in effecting the data transfer.

The nature of rule-based processing permits a loose organization of processing commands in that the execution is controlled by logic in rule sets with little regard to the location of the various rules. However, for design concept purposes, the rules were logically grouped by functional area. These areas were Examiner Rule Set, Judge Rule Set, Display Driver Rule Set and Executioner Rule Set as seen in Figure 3. The Examiner Rule Set is a logical group of rules that provides the initial, high level monitoring of anomalies. Rules triggered by these anomalies relay their occurrence by setting facts in the ART data base. The Judge Rule Set includes rules that react to these new facts and provide a more intensive analysis of the significance of these facts. This Judge Rule Set screens anomalies and discards benign conditions.

For example, the Examiner Rule Set might ascertain that certain engine measurements are out of tolerance. The Examine Rule Set conveys this information to the Judge Rule Set by introducing facts into the ART data base. The Judge Rule Set has the responsibility of using correlation algorithms to determine the validity of the transducer readings indicating the anomalies. If a valid anomalous condition is determined, the Judge Rule Set then introduces new facts into the ART data base as hard evidence of a problem. The Display Driver Rule Set then reacts to these new facts and provides the appropriate interaction with the domain expert who would be viewing the terminal attached to the Symbolics processor. The function of the Executioner Rule Set is to provide definite commands which should be acted upon based on the verified anomaly. For example, the Executioner Rule Set could set additional facts in the data base which would cause the Display Driver Rule Set to tell the domain expert to use voice communication with the pilot to change flight dynamics. The confidence in relying on computer-generated suggested corrective activities could be augmented by using the ART viewpoint analysis capability which would be implemented in conjunction with the Executioner Rule Set. This viewpoint analysis provides for the walk through of possible ramifications of the suggested corrective commands on a “what if” basis.

The two way communication of the Flavors Bus Link permits data to be sent back to the Gould processor from the Symbolics processor using memory reads and writes. In this fashion, the Expert System can easily output to flight test displays on the Gould processor.

DISTRIBUTED RULE SET PROCESSING

The ART system supports rule-based processing by the service of rule sets with respect to their reaction to data base facts with an inference engine. The inference engine reacts to the occurrence of a new fact in the data base by searching all the rules in the system to see if any are sensitive to this fact. This processing involves the resolution of conflicts set up when more than one rule set is affected. In real-time data acquisition systems, several thousand measurements might be monitored where many of the measurements would be changing rapidly at a rate of hundreds of times per second. The Symbolism 3670 computer is rated at about one-MIP for sequential processing. For LISP-based processing, it performs as well as a four-MIP sequential processing machine. The overhead of the inference engine and the resultant data base processing typically results in a processing rate of around 70 rule firings per second.

A separate IR&D study done at the same time as this study indicated that the data acquisition processing module in the Gould processor could easily absorb a portion of rule processing load without perturbing the acquisition loop, which has a budget of 50 microseconds per measurement for a given 20,000 sample per second required processing rate. By offloading a portion of the rule processing load to the host processor, the overall monitoring capability of the integrated system could be maximized. Thus, a mini-inference engine is implemented as an Alert Function (see Figure 4) in the acquisition loop. The Alert Function provides high level inferencing and signals the expert system in the Symbolics processor to focus in and provide intensive analysis of a selected group of anomalous measurements. This signaling is done via the Flavors Bus Link interface.

TEST ENVIRONMENT

This study resulted in the architecture shown in Figure 5. Data from the flight test vehicle is telemetered to a Telemetry Front End Processor. The front end processor synchronizes, decommutates and provides engineering unit conversion of the data and places the data in an acquisition buffer. The Acquisition Processor investigates the data and provides further processing, which includes the determination of whether the measurement is a subject for the Alert Function. The Acquisition Processor then calls the Alert Function to do high level inferencing with boolean and arithmetic algorithms. Results of this analysis are placed in a Current Value Table (CVT) which is in the memory shared by the expert system in the Symbolics processor. This, in effect, passes data over the Flavors Bus Link where it is processed by the expert system via the LISP Input/Output task.

Display processing provides both the Flight Test Engineer and the domain expert with the appropriate displays, considering the constraints of CPU availability in both processors. The domain expert has physical location access to displays driven by the Gould CPU which are tailored to that specific need.

EXTRACTING EXPERTISE

The knowledge engineer for health and welfare monitoring systems must extract the thought concepts of the experts in the form of reactions to observed anomalies. These concepts must be verbalized and formulated into rule sets and priorities of execution. Unfortunately, this process is not an easy one.

Consider an expert observing a multitude of observed physical activities in the form of sensor readings. The expert's brain assimilates the readings at a particular instant and integrates these readings with all happenings over the recent time period. False indications must be ascertained and discarded, and focus must be made on only pertinent facts. This type of observance can be envisioned by looking at the collection of graphs on Figure 6. This figure depicts a continuous recording of measurement readings over a time period for an engine test. This test is known as a snap throttle test which involves the pilot "snapping" the throttle forward very quickly. Fuel is provided to the engine which results in an increase of engine pressure. For this particular test and set of recordings, at a certain point, the pilot engages one of the afterburners. The "light off detector" provides an indication of the success of the firing of the first afterburner and is a signal of go ahead for the firing of the remaining afterburners, as seen on the figure. As observed on the figure, the test results in an engine stall, which is evidenced by the rapid drop of engine pressure. The pilot senses this drop and rapidly snaps the throttle back which results in a refiring of the engine.

The building of an expert system which can detect this engine stall involves the formulation of algorithms that effectively observe all indicators, just as would be done by the attendant expert that is looking at this data on the ground. First, the expert correlates all measurements such that erroneous data can be discarded. In the previously described expert system architecture, this is done by the Examiner Rule Set and the Judge Rule Set. Anomalies are noticed and analysis made as to the validity of the observed readings. Correlation analysis of dependent readings are made in order to ignore superflous data. The expert might require further access to measurements that were not being continually observed. This same logic must be entered into the logic of the expert system. The remaining activity by the expert would involve the formulation of prescribed responses to correct the situation.

EXPERT SYSTEM USAGE

During the process of investigating the performance aspects of expert system usage in real time data acquisition applications, an attempt was made to determine other areas of applicability of expert system techniques. Most of these areas focused on that set of operator automation activities.

a. Automated Setup. Telemetry processing setup is already automated for the most part. Data tables are downloaded into the telemetry front end processor and read back for verification. If substitutable equipment is found to fail, the operator can switch to alternate redundant equipment. An expert system could provide this capability automatically, with the additional feature of providing history accumulation to failed equipment by number. To implement this type of activity with an expert system, the operation would be observed for a prescribed period. Activities that were observed to be work intensive and repetitive would be the most obvious targets for automation. This activity is an ideal candidate for expert system involvement based on similar work done on other projects.

b. Automated Display Selection. The capability to automatically bring up displays for the operator is a very questionable and hotly debated activity. Consider the case where the operator spots a critical item on a display and begins to track it earnestly, all the while communicating with a pilot as to how to correct a maneuver. Then, without warning, the computer automatically switches displays on the operator. This type of scenario has precluded the automation of display selection with traditional automation programs. However, the potential of the depth of logic with an expert system provides the confidence of automating the display selection. The idea is to place enough logic in the expert system to exhaust all possibilities of perturbations that could be caused by such a switch. The state of the art for expert systems in this type of activity is a long way from providing a current implementation.

c. Automated Hardcopy. In conjunction with the automated display selection, the capability to have an expert system perform display hardcopies becomes feasible. The expert system would base this activity on values of data being monitored. Displays could automatically switched to alternate views just long enough for the hardcopy, and then be switched back. Also, by analyzing trends, more selectivity could be made in order to minimize hardcopy output of redundant data, or of data of no concern. This represents a viable area for current expert system involvement.

d. Open-ended Systems. By inserting an expert system into the system generation function, it could be used to intelligently configure a hardware and software, allowing ease of future expansion. The overhead of this activity would be high initially, but maturity of this endeavor would result in a cost effective implementation. This is certainly a noble cause and worthy of consideration. Activity here is expected over the next few years rather than during the near term.

In all the above capabilities, along with those oriented towards data monitoring, a very cautious approach has to be taken. The capability to easily add logic to expert systems allows corrections to be made when deficiencies are discovered. Many iterations of this approach would be required in order to take control of the system away from the operator and give it to the expert system. The ability to estimate the cost of such iterations is not currently available.

CONCLUSION

This study provided significant insight into the design of an expert system to assist the Flight Test engineer in the monitoring of real-time health and welfare data. Study results indicated that a local host processor, containing a mini-inference engine, may be used to allow the current state of expert system technology to be employed to solve today's telemetry problems. Finally, the extraction of expert knowledge from the expert today is a new frontier, and certainly more an art than a science.

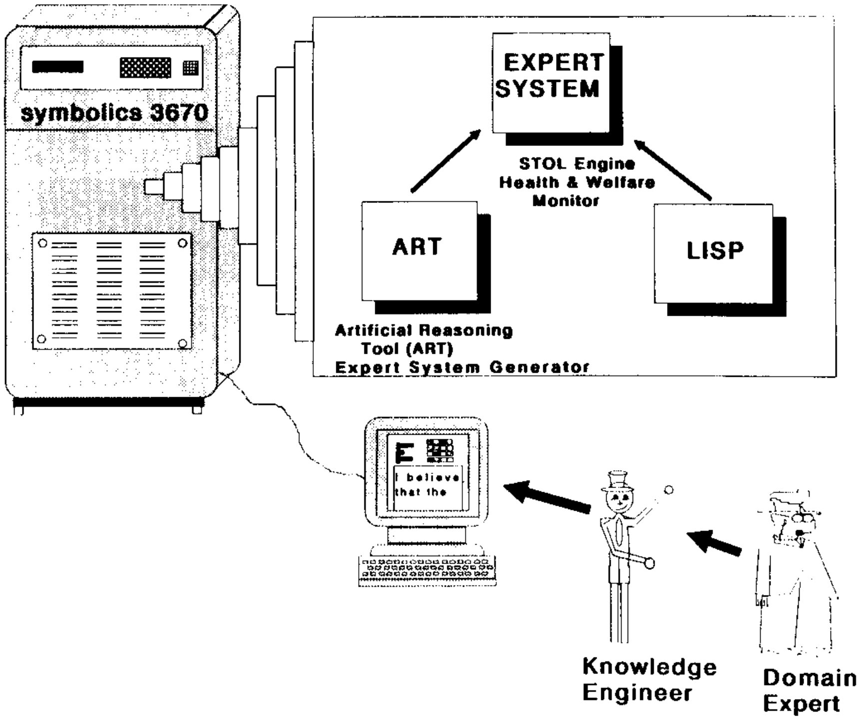


Figure 1. Development Environment (After Reference 1)

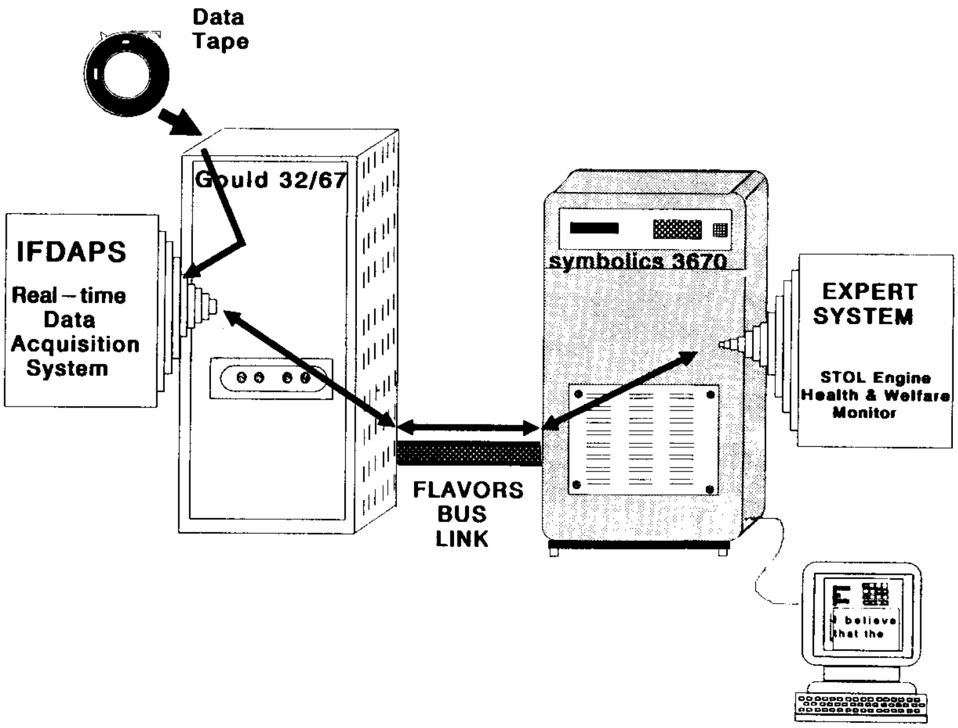


Figure 2. Validation Environment
(After Reference 2)

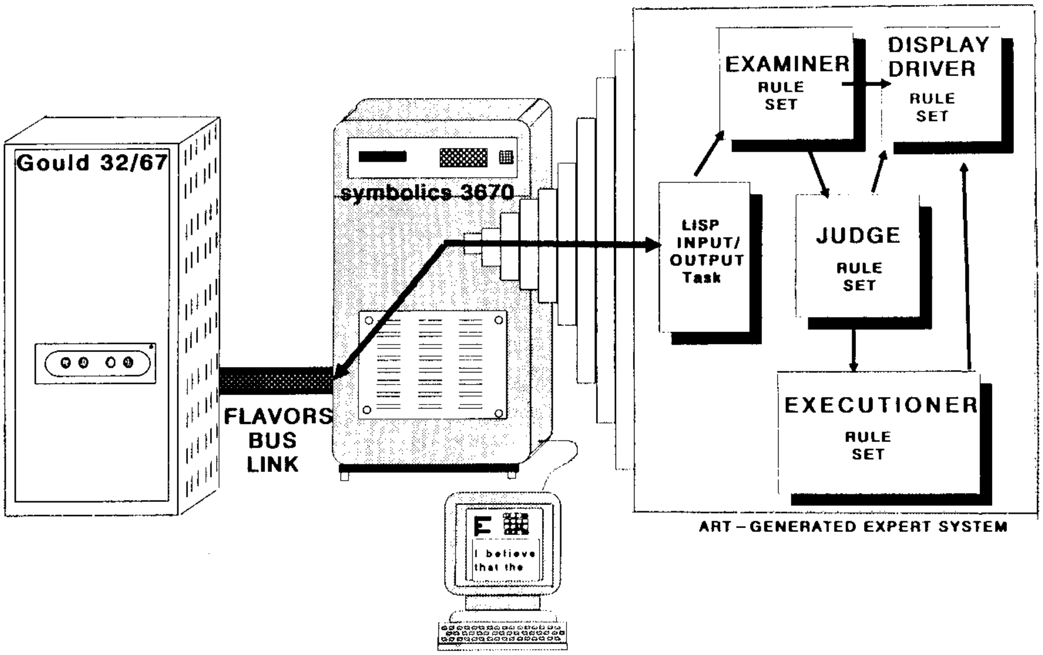


Figure 3. Expert System Architecture
(After Reference 3)

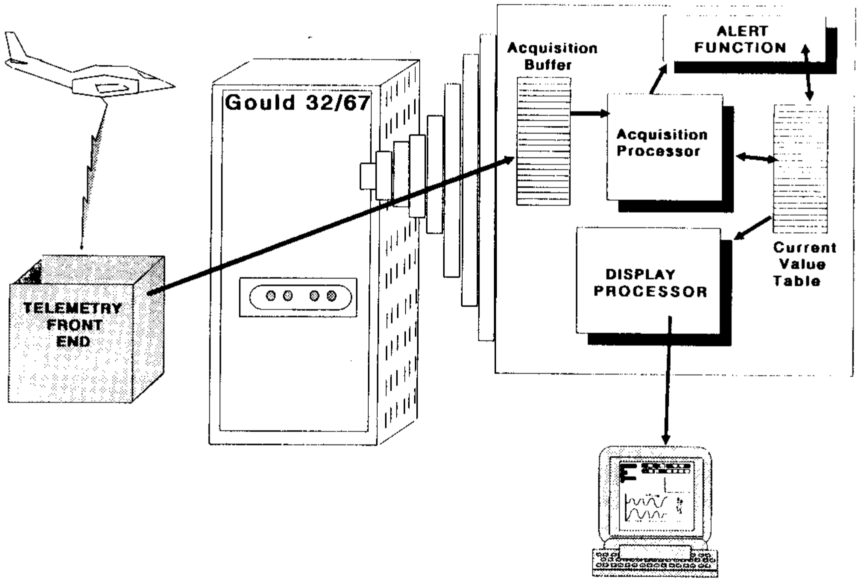


Figure 4. Current Flight Test Environment (After Reference 4)

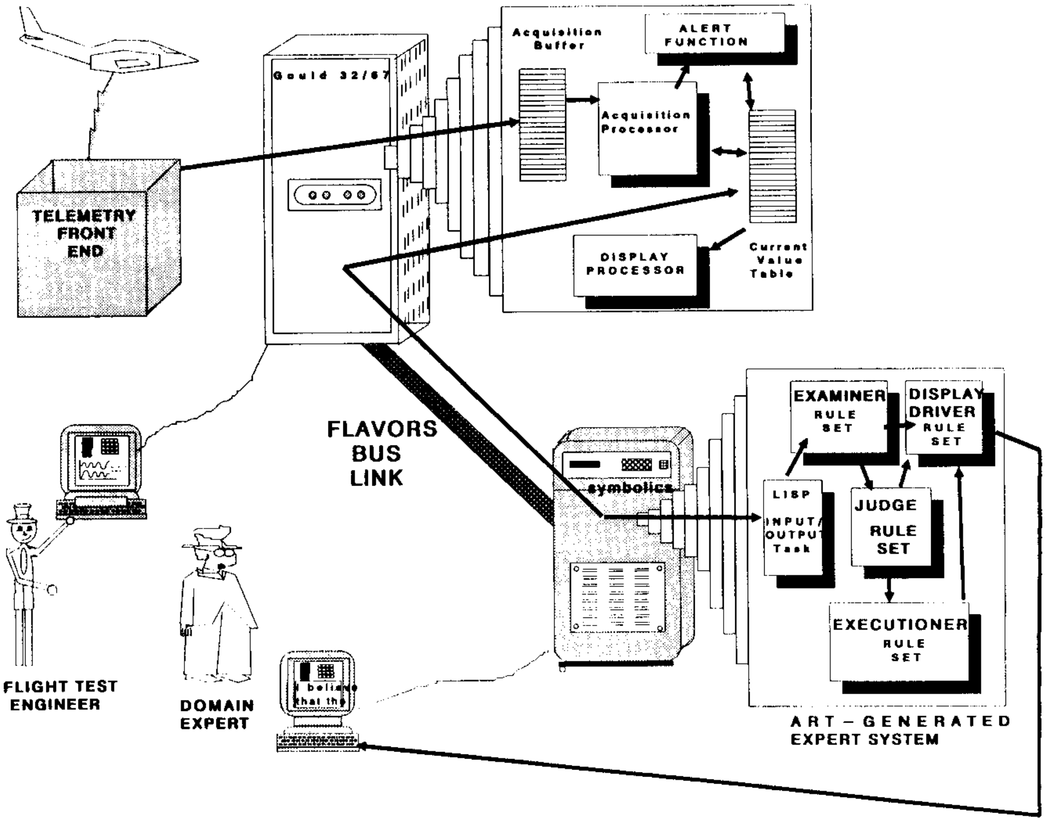
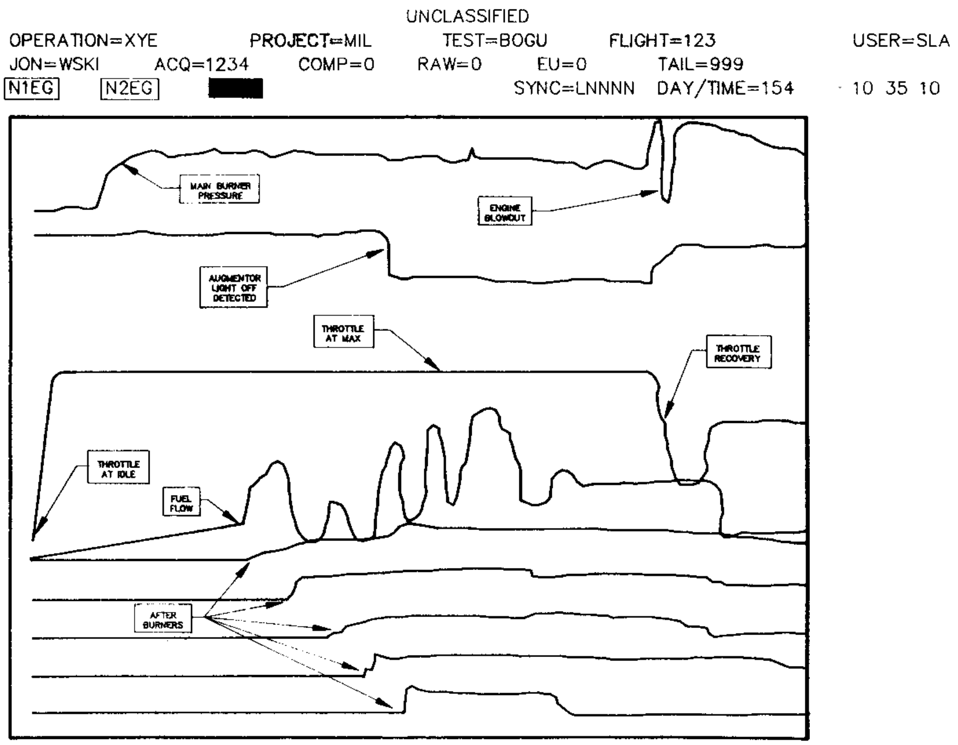


Figure 5. Real-Time Expert System Architecture (After Reference 5)



UNCLASSIFIED

OPERATION=XYE PROJECT=MIL TEST=BOGU FLIGHT=123 USER=SLA

JON=WSKI ACQ=1234 COMP=0 RAW=0 EU=0 TAIL=999

N1EG N2EG [redacted] SYNC=LNNNN DAY/TIME=154 10 35 10

Figure 6. A/C Engine Readout (After Reference 6)