PROTOTYPE FAULT ISOLATION EXPERT SYSTEM

FOR SPACECRAFT CONTROL

WALT TRUSZKOWSKI

FY84 RTOP 506-54-66

OBJECTIVE:

1. IDENTIFY SPACECRAFT OPERATIONS FUNCTIONS AT GSFC THAT COULD POTENTIALLY BE AUTOMATED USING AI TECHNIQUES.

2. DEVELOP A PROTOTYPE SYSTEM TO:

   - DEMONSTRATE AND VALIDATE EXPERT SYSTEM OPERATION FOR CONTROL CENTER SYSTEM DEVELOPERS AND OPERATORS.

   - PROVIDE EXPERIENCE FOR OPERATIONAL SYSTEM DEVELOPMENT.

PROTOTYPE FAULT ISOLATION EXPERT SYSTEM FOR SPACECRAFT CONTROL

W. TRUSZKOWSKI

ABSTRACT

Over the past year, work has been ongoing to identify areas of spacecraft command/control which could benefit from automation using artificial intelligence (AI), especially expert systems technology. A major part of this work has been the development of a demonstration expert system to help illustrate system attributes and development methodology.

An output of this activity is a program which illustrates how an expert system might perform fault handling for a satellite propulsion subsystem. The program runs on a VAX 11/780 under VMS with a VT100 interface and is written in FRANZ/LISP. The Program presents a series of menus to the user, requesting the selection of:

- The level of explanation which is presented to the user
- The displaying of assertions (fault symptoms passed to the rule base)
- The displaying of rules which "fire" during the running of the program
- The choice of Hydrazine Propulsion Subsystem (HPS) configuration (currently either ISEE or IUE configurations are selectable)
- The choice of fault type and fault location

Once the user make these selections, the program simulates the behavior of the HPS under the fault condition, detects the presence of the fault, isolates the exact nature and location of the fault, and then determines what set of actions to perform in implementing the fault workaround.

During the running of the program, a graphical depiction of the chosen HPS configuration is displayed, occupying all but the bottom five lines of the screen. The remaining portion of the screen is used to display explanations of faults and workaround actions. The display of the HPS includes the telemetry values for the sensors present in the system, which are updated during the running of the program.

This activity has provided insight into the use of expert system technology in the spacecraft ground command/control environment. This prototype system will now be used to demonstrate and evaluate the use of expert systems in a control center environment. The propulsion system simulation and display approach should allow control center operators to experiment with and understand the operation of expert systems and gain confidence in their operational use.

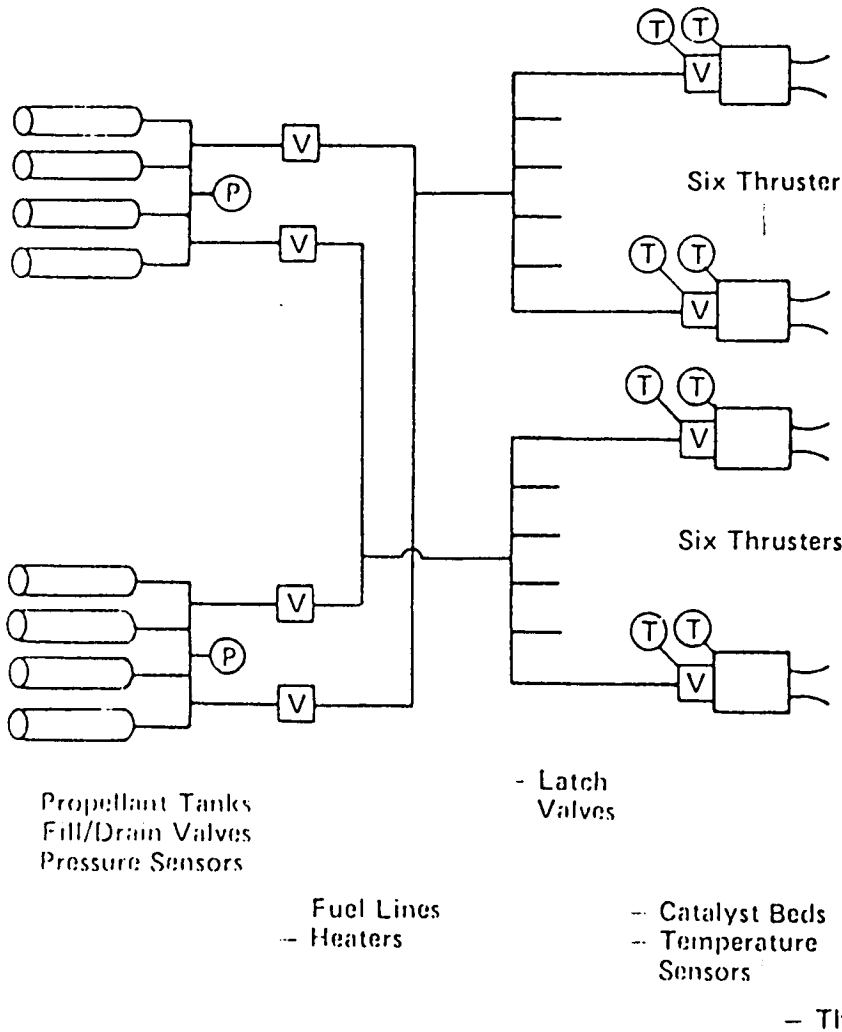
PROTOTYPE FAULT ISOLATION EXPERT SYSTEM

FOR SPACECRAFT CONTROL

PROTOTYPE SYSTEM

- o DEVELOPED BY MARTIN-MARIETTA/DENVER
- o EXPERT SYSTEM FOR FAULT ISOLATION/RECOVERY IN HYDRAZINE PROPULSION SYSTEM OPERATION
- o MODELS AND GRAPHICALLY DISPLAYS OPERATION OF ISEE-1 AND IUE PROPULSION SYSTEMS
  - PHYSICAL MODEL (IDEALIZED)
  - TELEMETRY SIMULATOR
- o ALLOWS USER INTRODUCTION OF MULTIPLE FAULTS
- o DISPLAYS RULES AND WORK AROUND PROCEDURE
- o IMPLEMENTATION: FRANZ/LISP AND MRS

ISEE Hydrazine Propulsion System



Fault Detection
- Exposure module performs "unmasking" activity.
- Telemetry generator produces telemetry data & reports differences from expected valves.

Fault Isolation
- Fault handler acts on telemetry generator data.
- Fault handler generates hypotheses and tests by commanding the S/C.

Fault Workaround
- Isolates failed component.
- Allows limited operation.
- Maintains workaround during second fault analysis.

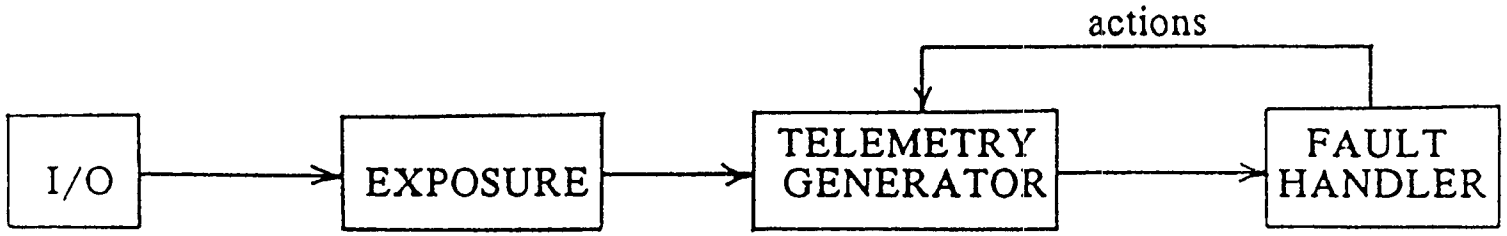
Note:

1. Fault handling module does not have an explicit model of the HPS.
2. Fault handler input comes exclusively from the telemetry generator (via the executive).

- *HPS Demonstration System provides high degree of visibility into Expert System activity:*
  - *symptom generated assertions*
  - *binding variables to LHS of rules*
  - *RHS procedures*
  - *rule generated assertions*
  - *forward chaining process*
- *HPS Demonstration System provides insight into Expert System application potential for:*
  - *fault handling/troubleshooting*
  - *executive controller for automated center*
  - *scheduling*
  - *mission workarounds*

FAULTS TO BE CONSIDERED:

- *Leak in the hydrazine storage tank*
- *Leak in a section of the line*
- *Failure of a latch valve to open*
- *Failure of a latch valve to close*
- *Failure of an engine valve to open*
- *Failure of an engine valve to close*
- *Failure of a heater to turn off*
- *Failure of a heater to turn on*
- *Failure of a catalyst bed*
- *Sensor failures*
  (engine and latch valves [open/closed], fuel pressure sensors, fuel line and catalyst bed sensors [on/off], fuel line and catalyst bed temperature sensors)



- prompts user for input:
  - configuration
  - fault type
  - fault place
- status
- explanation

- determines routine HPS actions so fault causes abnormal behavior

- executes actions, calculates telemetry values for new state, reports abnormal data

- isolation procedures
- workaround procedures
- explanation procedures
- knowledge base

CONCEPTUAL FLOW

- *USER SPECIFIES TYPE AND LOCATION OF FAULT*
- *SYSTEM MODIFIES INTERNAL MODEL ACCORDINGLY*
- *TELEMETRY GENERATION COMPONENT PRODUCES APPROPRIATE TELEMETRY DATA*
- *FAULT HANDLING COMPONENT EXAMINES TELEMETRY DATA AND USES ITS RULE BASE AND KNOWLEDGE OF NOMINAL STRUCTURE AND FUNCTION TO LOCATE FAULT (MAY REQUIRE "COMMANDING" THE MODEL)*
- *FAULT HANDLING MODULE USES RULE BASE TO IDENTIFY WORKAROUND*
- *MODEL IS MODIFIED AND TELEMETRY NORMALIZES*

KNOWLEDGE BASE ARCHITECTURE

- STRUCTURE OF TYPICAL RULES

```
(if (and ( <find-assertion> )
         ...
         ( <find-assertion> )
         (unknown ( <find-assertion> ))
         ...
         (unknown ( <find-assertion> )))
    (runnable (assert-and (and ( <lisp-procedure> )
                               ...
                               ( <lisp-procedure> )
                               ( <make-assertion> )
                               ...
                               ( <make-assertion> )))))
```

- KNOWLEDGE BASE IS PARTITIONED
  - GLOBAL AND SPECIFIED "THEORIES"

RULE FORMAT:

Telemetry Generator produces a list of values that are fault symptoms in a general format:

(|component| |number| |qualitative value| |telemetry| |expected|)

as variables:

(|$c| |$n| |$q| |$t| |$e|)

which are matched to rules:

```
(IF (and (assertion is true)
         (constraint is not true))

(THEN (and (LISP procedure to print rule)
           (LISP procedure to print explanation)
           (place data/result in working memory))))
```

PROTOTYPE FAULT ISOLATION EXPERT SYSTEM FOR SPACECRAFT CONTROL

STATUS

- o PROTOTYPE INSTALLED AT GSFC
- o INITIAL DEMONSTRATIONS TO CONTROL CENTER OPERATIONS MANAGERS

NEXT STEP

- o EVALUATION BY SPACECRAFT CONTROLLERS
- o EVALUATE SYSTEM FOR EXTENSION TO OTHER SPACECRAFT AND/OR SUBSYSTEMS