TRANSITION FLIGHT CONTROL ROOM AUTOMATION

by
Curtis Ray Welborn
National Aeronautics and Space Administration,
Lyndon B. Johnson Space Center
Mail Code FS72
NASA Road 1
Houston, Texas 77258

TRANSITION FLIGHT CONTROL ROOM AUTOMATION

Curtis Ray Welborn
NASA/Johnson Space Center

February 1990

ABSTRACT

The Workstation Prototype Laboratory is currently working on a number of projects which we feel can have a direct impact on ground operations automation. These projects include:

• The Fuel Cell Monitoring System (FCMS), which will monitor and detect problems with the fuel cells on the Shuttle. FCMS will use a combination of rules (forward/backward) and multi-threaded procedures which run concurrently with the rules, to implement the malfunction algorithms of the EGIL flight controllers. The combination of rule based reasoning and procedural reasoning allows us to more easily map the malfunction algorithms into a real-time system implementation.

• A graphical computation language (AGCOMPL). AGCOMPL is an experimental prototype to determine the benefits and drawbacks of using a graphical language to design computations (algorithms) to work on Shuttle or Space Station telemetry and trajectory data.

• The design of a system which will allow a model of an electrical system, including telemetry sensors, to be configured on the screen graphically using previously defined electrical icons. This electrical model would then be used to generate rules and procedures for detecting malfunctions in the electrical components of the model.

• A generic message management (GMM) system. GMM is being designed as a message management system for real-time applications which send advisory messages to a user. The primary purpose of GMM is to reduce the risk of overloading a user with information when multiple failures occurs and in assisting the developer in devising an explanation facility.

The emphasis of our work is to develop practical tools and techniques, while determining the feasibility of a given approach, including identification of appropriate software tools to support research, application and tool building activities.

AGENDA

INTRODUCTION

FISCAL YEAR 1989

FISCAL YEAR 1990

ARMOA PROJECT

- Mission Operations Support Study (MOSS)
- Alternate Language Interface (ALI)
- Generic Message Management (GMM)

INTRODUCTION

This is a Code S RTOP sponsored by Gregg Swietek from NASA Headquarters. The work for the Transition Flight Control Room has been conducted by the Workstation & Visual Systems Branch which is part of the Systems Development Division. In fiscal year 1989 the team members were Allen Brewer, (NASA/Section Head of the Workstation Systems Development Section), Clark Pounds, (NASA/Lab. manager for the Workstation Prototype Laboratory), Danny Labasse (MITRE) and Dave Hammen (MITRE). For fiscal year 1990 team members are Allen Brewer, Curtis Welborn (NASA), Frederic Gibbs (NASA), Charlie Robertson (McDonnell Douglas), Wayne Parrott (LinCom) and Yashvant Jani (LinCom). The objectives of the Transition Flight Control Room are characterized in the following paragraphs.

> At some point in the prototyping process, it is necessary to test the software using operational data. Such testing is difficult in an operational environment characterized by strict controls that permit only qualified software to execute. In a near-operational environment, in which some of the strict controls are removed, near-operational data can be fed to prototype software and aid the prototyping process. The Transition Flight Control Room (TFCR) provides such an environment, allowing control center prototypes to be tested using operational data. NASA's Hardware Independent Software Environment (HISE) provides standardized tools and rules for software developed for the TFCR and related workstation laboratories, but does not yet provide support for advanced automation techniques.
>
> The goal of the TFCR advanced automation task is to augment the HISE with appropriate tools and techniques so that advanced automation software may be developed within the HISE and used in the TFCR.[1]

INTRODUCTION

- Test Flight Control Room Automation Task (TFCR)
  Code S RTOP : Gregg Swietek

- JSC MSD/SDD/Workstation & Visual Systems Branch/
  Workstation Systems Development Section (FS72)

- FY 1989 Members
  (NASA) Allen Brewer (Section Head), Clark Pounds
  (MITRE) Daniel Labasse, David Hammen
  (Ford Aerospace) John Engvall, Matt Hanson, Charles Copeland

- FY 1990 Members
  (NASA) Allen Brewer, Curtis Ray Welborn, Frederic Gibbs
  (LinCom) Wayne Parrott, Yashvant Jani
  (McDonnell Douglas) Charlie Robertson

TFCR's goal is to augment NASA's Hardware Independent Software Environment with tools and techniques which can be used in automating the control center environments (SSCC/MCC)

Fiscal Year 1989

For most of 1989 the MITRE Corporation conducted interviews and worked to produce a report describing functional requirements, system requirements, and selection factors for the advanced automation tools to be acquired for the TFCR/HISE. The addition of 7 methodologies into the TFCR were recommended in the final MITRE report. The 7 methodologies were : Rule-Based Reasoning, Hypermedia, Object-Oriented Programming, Model-Based Reasoning, Databases, Voice Generation and Computer-Aided Software Engineering Tools. In addition to the 7 methodologies recommended by MITRE two methodologies, Neural Networks and Analogical (Case-Based) Reasoning, were mentioned as future additions to the TFCR.

The current members of this RTOP strongly suggest the addition of some form of Procedure-Based Reasoning to augment existing Rule-Based Reasoning systems. Our desire for Procedure-Based Reasoning is driven by the need to execute **multiple** procedural algorithms (e.g. malfunction procedures) concurrently within a Rule-Base environment. Because of the need to share information between procedural algorithms and rules, an environment which intergrates both rules and procedures offers the best development and maintaince environments. This ability would allow concurrent execution of any number of algorithms while continuing to perform data driven monitoring of the systems' health and status.

FISCAL YEAR 1989

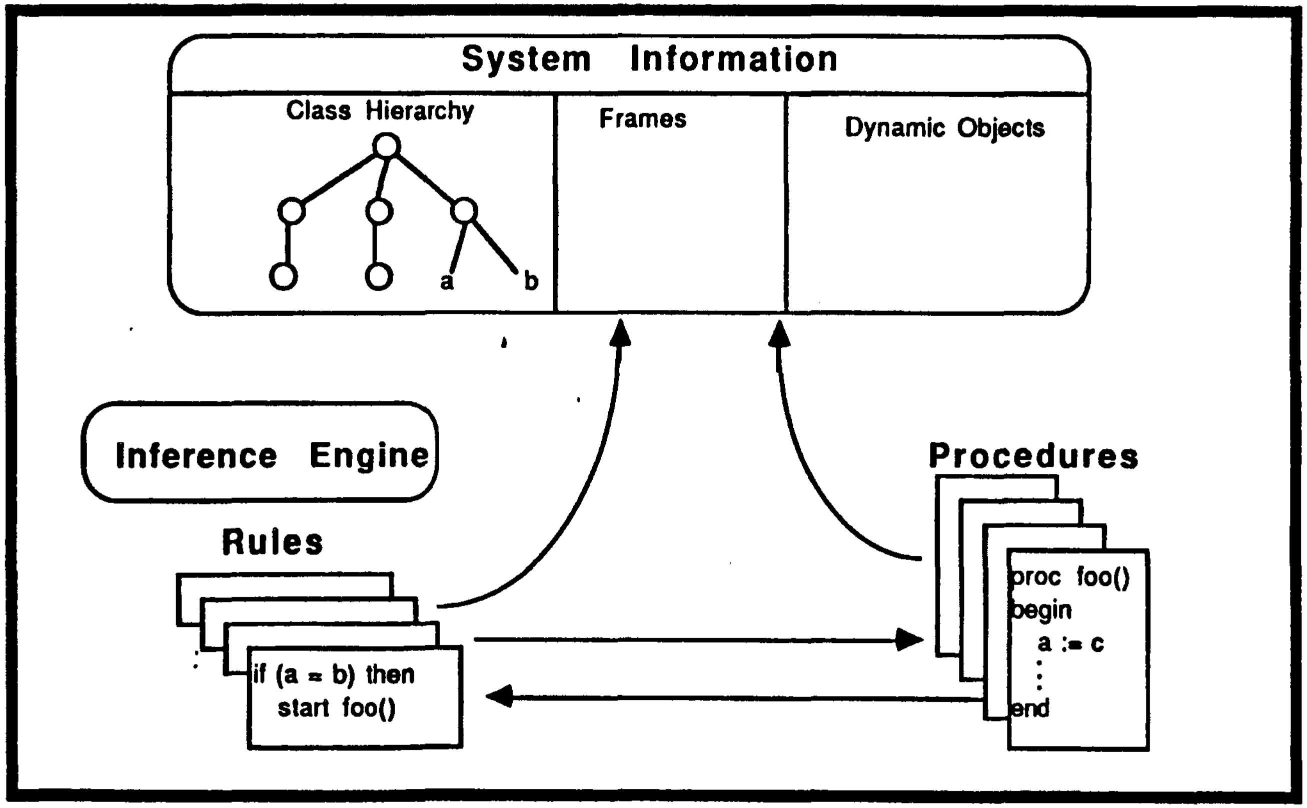
•MITRE Final Report

••Recommended Methodologies to the TFCR
Rule-Based Reasoning
Hypermedia
Object-Oriented Programming
Model-Based Reasoning
Databases
Voice Generation
Computer-aided Software Engineering Tools

••Functional Requirements
Developed for Recommended Methodologies

FISCAL YEAR 1989 (continued)

- •Additional Methodologies
  - Neural Networks
  - Analogical (Case-Based) Reasoning
  - Procedure-Based Reasoning



```
if (a = b) then
 start foo()
```

```
proc foo()
begin
  a := c
  :
end
```

Fiscal Year 1990 Work

As of this fiscal year a new project within the TFCR task has started, Applied Research in Mission Operation Automation (ARMOA) task. The objectives of this project are to evaluate and construct software systems which will aid in the automation or documentation of process within the control centers and training facilities. This is to be accomplished by automating and/or easing the acquisition of knowledge, the design of knowledge structures and the development, validation and improvement of expert systems. Three different project areas currently exist within ARMOA: MOSS, ALI, and GMM. While the projects are distinct in nature, there is a great deal of information shared between the projects. The MOSS project, due to its operational nature, is currently being used tσ direct most of the research and implementations in our other projects.

MOSS

MOSS, the Mission Operations Support Study, is a development project in which an operational system will be developed and studied. By studying both the technical and human issues within the current control center, we are better able to direct our research in how we can construct new systems for the Space Station. The major activity of MOSS is currently the construction of the Fuel Cell Monitoring System (FCMS), a health and status monitoring system for the EGIL flight controllers. The system is being directed at, but not limited to, monitoring and detecting problems with the Fuel Cells onboard the Orbiter. By actively pursuing the development of an operational system, while studying the operational environment it must work within, we are gaining valuable insights into what is needed and what is wanted. Two major areas of study exist for us while developing FCMS: Environmental Studies and Technology Evaluation.

In our Environmental Studies we are most interested with how the current job of monitoring gets done and what the users (flight controllers) want in new systems. This involves understanding issues related to:

- operating and constructing real-time health and status monitoring systems;
- dealing with long duration monitoring in the face of computer system failures, the reconfigurations of equipment due to physical changes or break downs;
- technology transfer issues such as user interfacing, implementing active vs. passive resource management systems, and the development of user trust for these new technologies.

Technology evaluation is our second primary area of interest. We are currently evaluating the use of G2, a real-time expert system development environment and the relationship between rules and algorithms for implementing an operational system.

FISCAL YEAR 1990

•Applied Research in Mission Operation Automation (ARMOA)

••Mission Operations Support Study (MOSS)

•••Operational Project

EGIL Fuel Cell Monitoring System (FCMS)

•••Environmental Studies

Real-time Health & Status Monitoring

Long Duration Health & Status Monitoring

System Failure Recoveries

Reconfiguration Dynamics

Technology Transfer

Passive/Active System Management

Interface Designs

•••Technology Evaluation

G2

Rules vs. Procedures

ALI

ALI (Alternate Language Interface) is our second major project. The three subprojects currently being worked are centered around the basic theme of capturing knowledge from a user by supplying the user with languages which allow them to encode their knowledge more easily. These subprojects are:

- ATA (Alarms Triggers & Algorithms) is a software architecture and design philosophy which is being defined for real-time monitoring systems. Syntax and semantics to support ATA are also being defined, though we have no plans to implement this architecture. We would prefer, when finished with the architecture, to present it to existing commercial developers for inclusion into their products. ATA has been, and is continuing to be, heavily influenced by the Procedural Reasoning System (PRS) from SRI International and G2 from Gensym Corp. Both G2 and PRS support a form of Procedure-Based Reasoning though their implementations differ greatly.

- AGCOMPL (A Graphical Comp Builder) is a graphical programming prototype that produces source code for the MSD COMP Builder using the extended MOLE grammar. The graphical language design of AGCOMPL was derived from circuit diagrams, where every operation to be performed has a unique icon, (e.g. AND GATE and OR GATE in circuit diagrams vs. EQUAL GATE and ADD GATE in AGCOMPL). AGCOMPL has been designed for non-programmers or people with little programming experience. Phase 1 of AGCOMPL will be completed this month, with an evaluation process to follow. Commercial products which support this same form of graphical programming are being reviewed in the Workstation Prototype Laboratory.

ALI, continued.

- MEPS (Modeling of Electrical Power Systems) is one of a set of projects relating to the modeling of physical devices. The overall objective of these projects is to provide a modeling environment capable of producing rules or algorithms for detecting failures at the Orbital Replacement Unit (ORU) level. The generation of rules or algorithms from our models is to be implemented on two separate levels of reasoning. Level 1 is to use qualitative reasoning based on the relationship of how a model's ORU components are connected. Level 2 is to use quantitative measures generated by the modeling of the physics of the system at the ORU level. Simulators for each ORU will be used at this level if they exist.

MEPS is the first of our set of modeling projects. MEPS is to be developed over a number of phases, with the ORU components of an electrical system being represented by icons and the physics of the system being encoded into rules and algorithms. Electrical components for which icons and modeling capabilities will exist are sensors, power sources, breakers/fuses, switches and loads at the ORU level. Additionally the ability to introduce a failure into the model will be provided. Later phases of MEPS call for the modeling of mechanical components which regulate flows and pressures and the sensors that measure these levels. To generate algorithms for the detection of failures, sensors (e.g. voltage, current, pressure, flow rate, status) must be present in the model.

FISCAL YEAR 1990 (continued)

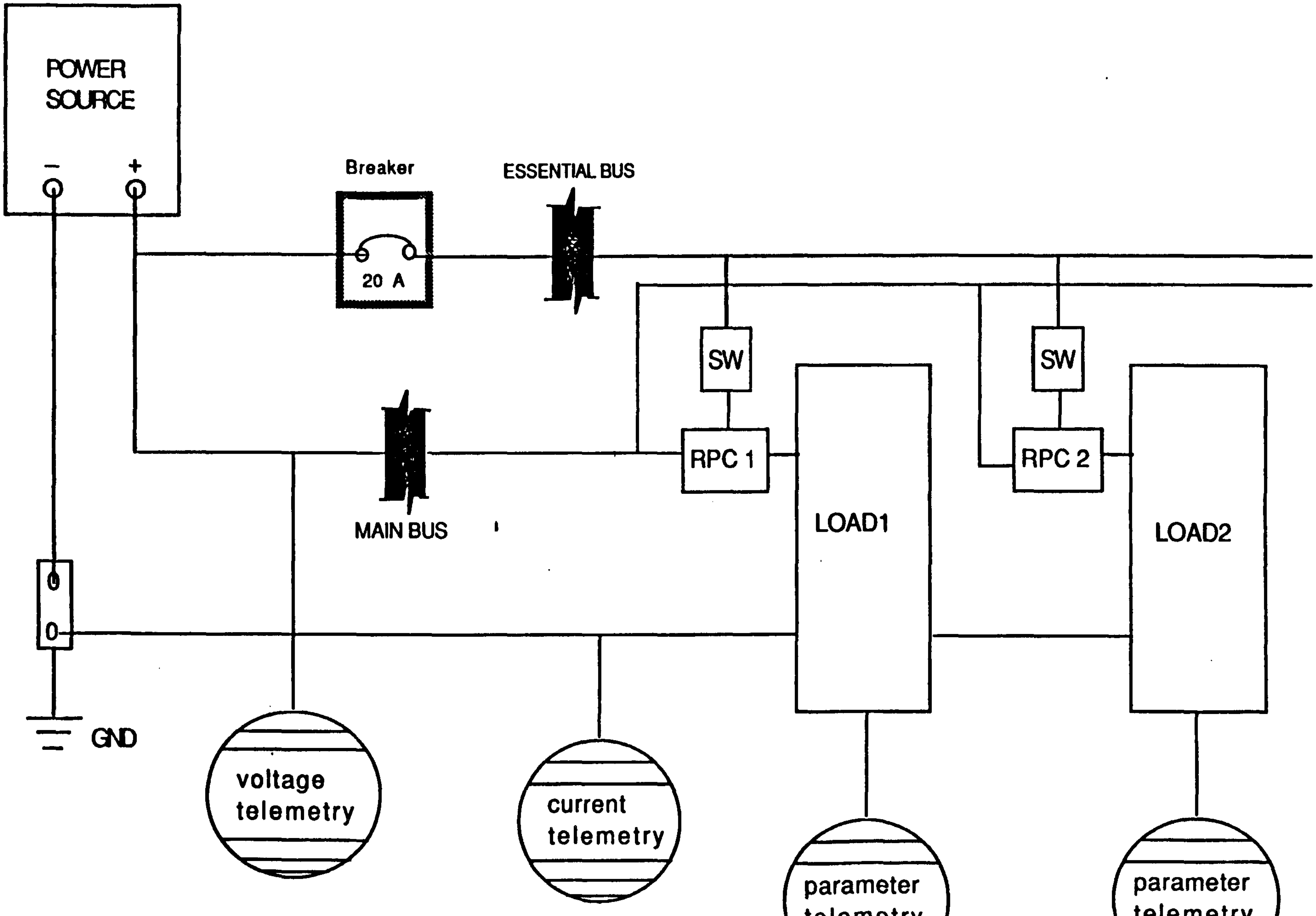
••Alternate Language Interface (ALI)

•••Alarms Triggers & Algorithms (ATA)
Software Architecture

•••A Graphical Comp Builder (AGCOMPL)
Graphical Language for Designing Comps
Language Design derived from Circuit Diagrams
Graphical Language meant for non-programmers

•••Modeling Electrical Power Systems (MEPS)
Model Electrical Subsystems/Components
Model Electrical Component Failures
Generate Failure Detection Algorithms Using
Qualitative Reasoning and Quantitative Measures

•••Model Mechanical Components (MMC)
Model Selected Mechanical Components
Introduce Mechanical Components into MEPS



RPC : REMOTE POWER CONTROLLER

SW : SWITCH

GMM

Generic Message Management (GMM) is the final project in the ARMOA task. GMM is being designed to provide various message management capabilities for advisory messages sent to a user. While the design of GMM is intended for use with real-time monitoring systems, any system which would require message management could benefit from GMM. There are three functional units to GMM :

- Display Management (DM) - Will manage the screen resources and filter messages using qualitative and time dependency priorities.
- Review Management (RM) - Will manage the reviewing of previous messages and the reviewing of user defined message hierarchies.
- Erase Management (EM) - Will manage the removal of messages from a user defined message hierarchy as well as logging of removed messages for later analysis.

FISCAL YEAR 1990 (continued)

••Generic Message Management (GMM)

•••Display Management (DM)
Qualitative and Time Dependent Priorities
Filter Messages
Manage Screen Resources

•••Review Management (RM)
Review Previous Messages
Review Dependent Message Hierarchies

•••Erase Management (EM)
Log Message to Disk as Removed
Remove Messages from Internal Hierarchies