N89-19830

ARTIFICIAL INTELLIGENT DECISION SUPPORT FOR LOW-COST LAUNCH VEHICLE INTEGRATED MISSION OPERATIONS

Dr. Gerard P. Szatkowski
General Dynamics / Space Systems Division
5001 Kearny Villa Rd., San Diego, Ca. 92138

Dr. Roger Schultz
Abacus Programming Corporation
14545 Victory Blvd., Van Nuys, Ca. 91411

Sponsored by: Flight Dynamics Lab., AFWAL WPAFB

ABSTRACT

This paper reviews progress in a current study assessing the feasibility, benefits, and risks associated with AI Expert Systems applied to low cost space expendable launch vehicle systems.

This study is in support of the joint USAF/NASA effort to define the next generation of a heavy-lift Advanced Launch System (ALS) which will provide economical and routine access to space. The significant technical goals of the ALS program includes: a 10 fold reduction in cost per pound to orbit, launch processing in under 3 weeks, and higher reliability & safety standards than current expendables.

General Dynamics with Abacus, under the auspice of Wright Aeronautical Labs WPAFB, are exploring the use of knowledge-based system techniques. This is for the purpose of automating decision support processes in on-board and ground systems for pre-launch checkout and in-flight operations. Issues such as: satisfying real-time requirements, providing safety validation, hardware & DBMS interfacing, system synergistic effects, human interfaces, and ease of maintainability, have an effect on the viability of expert systems as a useful tool.

INTRODUCTION

The Problem — We recognize that our nation's current suite of launch vehicle systems has a number of problems making them inadequate for the projected needs after the mid-1990's. High costs of above $2,000/lb of payload delivery, a low reliability, poor resiliency (standdowns of many months for current expendables), and limited launch rate capacity are reasons behind the joint USAF/NASA effort for an operational ALS and Shuttle II. These will serve the commercial and DoD mission models beginning in 1995. If we are to meet the goals of $300/lb and launch rates as high as 50 missions annually, these systems and their associated ground operations segment must be made as autonomous as possible, while at the same time improving reliability and safety. This study explores the use of knowledge-based system (KBS) techniques for the purpose of automating the decision processes of these vehicles and all phases of the ground operations segment by assessing the feasibility, benefits, and risks involved.

An expert decision aid is a software approach to solving particular problems that are constantly changing over time and are complex or adaptive in behavior, the opposite of an analytical problem that is basically deterministic. Examples of these types of problems are: the re-scheduling of a vehicle checkout due to a damaged cable; or, determining if a system is indeed faulty given conflicting sensor readings. These heuristic problems require a depth of knowledge and experience (art rather than science) to form solutions quickly. Expert systems embody that collection of knowledge and experience in modular pieces that are rules and facts that describe the proper thought process for a given set of circumstances arrived at by any path. It is this modular independence that makes expert systems attractive. The incremental improvement of knowledge and experience can be built and tested readily without re-testing the rest of the software system, unlike conventional software that is difficult to maintain in a day to day changing environment.

Project Scope — The scope of the Space Transportation Expert System Study, (STRESS) is extensive. Virtually all on-board management decision functions are included as is the ground segment from pre-mission planning, through checkout and launch services, and post-flight analysis. This effort is then to define viable Vehicle Mission Management System (VMMS) architectures (and their corresponding ground system) which can evolve with this new KBS technology, determine the most fruitful to pursue, and for specific areas derive the avenues in need of concentration. This process is depicted in Figure 1. It is clear that the focus of our efforts must be on the AI techniques supported with the knowledge of the functional systems requirements. Figure 2. shows a rough overview of the ALS program major milestones and the expected implementation of KBS as a technology.

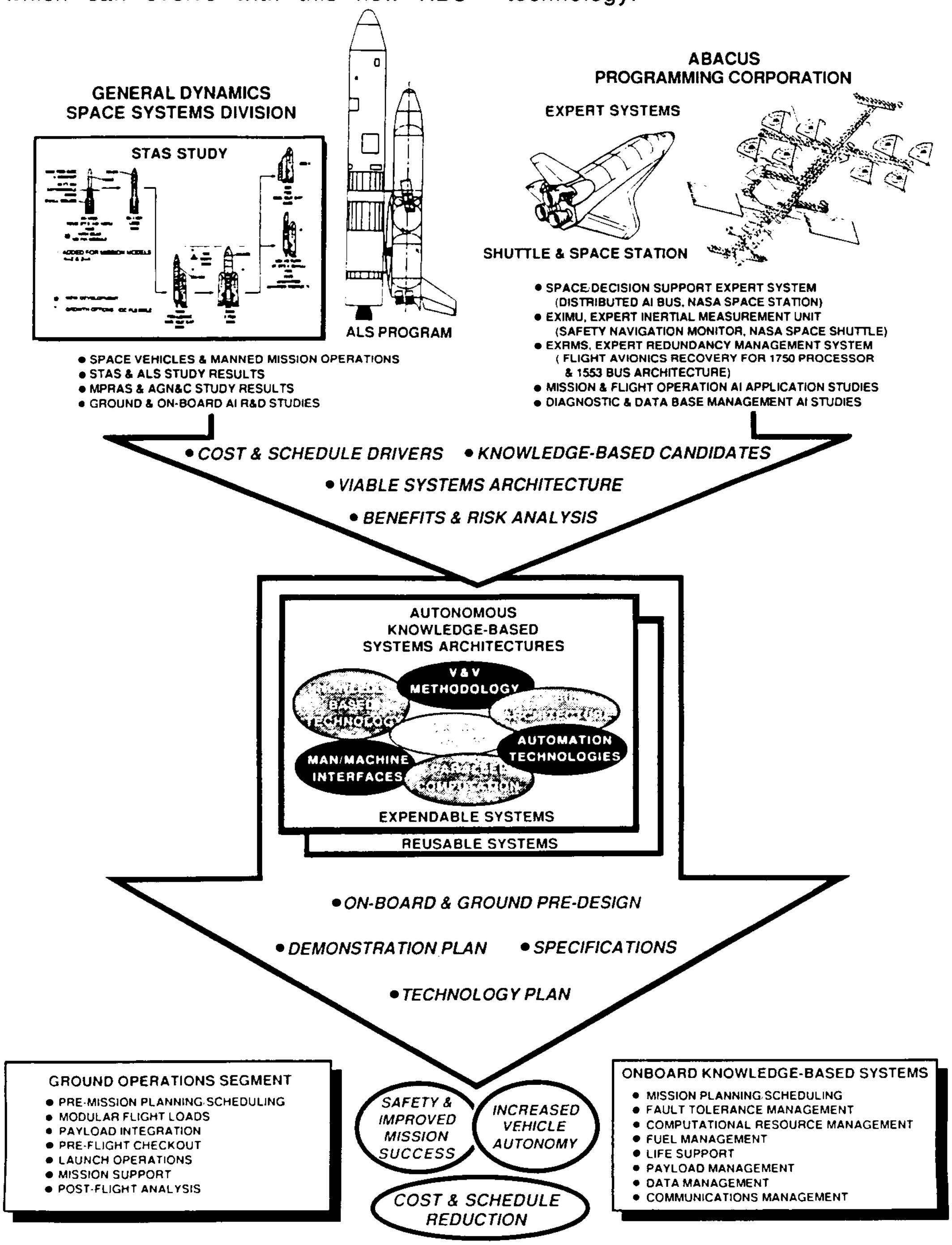

Figure 1. The Space Transportation Expert System Study, STRESS

ORIGINAL PAGE IS
OF POOR QUALITY

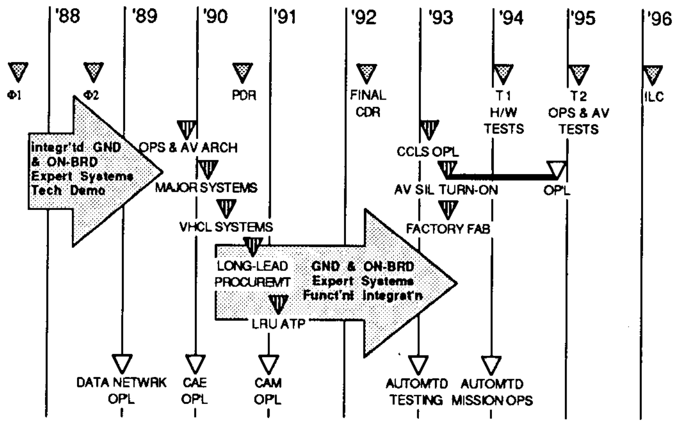

Figure 2. ALS Program Expert System Integration

Problems related to AI applications we will assess include:

- The increasing cost of failure, which leads to a high demand for success and hence system reliability
- Ever increasing system demands, i.e., reconfigurable avionics, adaptable GN&C, heavy instrumentation, and autonomous procedures like rendezvous and docking, etc.
- Multiple payloads in heavy lift vehicles which tend to complicate integration, planning requirements, and V&V of flight loads, and
- Increasing flight rates, which tend to magnify the logistical dependence on ground support; Problems related to AI applications we will access include:
- Speed, where the current technology is not well suited to real-time situations
- V&V methodology, which has been inadequately developed to date
- Large data storage requirements, which complicates redundancy
- Knowledge representation and manipulation techniques where, to date, no standard exists and where optimized hardware has been slow to develop

To attack these problems, we must rely on technological innovations. Based on our studies, we believe the key to viable solutions must address improving speed and modularization of the multiple support functions which comprise the system. To accomplish this, two approaches we are considering are: the utilization of parallel processor techniques with a tiered management approach for modularized expert systems; and utilization of a distributed AI bus, developed for the space station, to network the functional systems. The basis of our approach is to conceptualize systems employing varying combinations of these philosophies.

BENEFITS AND LIMITATIONS OF KBS

Although there are many considerations that will be made in the application of KBS to Launch Systems, some of the major issues are:

Benefits—

- The major benefit of knowledge based languages and techniques is the obvious one that they allow the modeling and solution of problems that are inordinately difficult using conventional procedural programming.
- Ease of maintenance results from knowledge being expressed in small independent units without the complex interrelationships that result from the sequence-specific nature of procedural code.
- Ease of top-down representation and development also results from having knowledge expressed in independent units. Often it is possible to define major aspects of a problem at a relatively abstract level and gradually elaborate the knowledge to deal with more and more specific issues, in the same way an expert develops expertise. Knowledge developed in this way can be much easier to understand and verify than procedural knowledge.
- A corollary of the above advantages is that systems can be built to perform reasonably in combinations of circumstances that are unanticipated by the designer. However, there is the corresponding danger that a system will act when it should not.

Limitations—

- Real-time software applications face all the same challenges as other software applications, and in addition are faced with the problem of synchronizing with real world events. Developing a knowledge based system for real-time performance is a challenge best met by designing the performance in from the start, although tuning and hardware upgrades can have a real impact in certain cases. Design for real-time should follow two principles: 1) locate processing nodes to limit data transfer to the most essential data to avoid bottlenecks; and 2) consider distributing or pipe line processing over a number or processors.
- There is a shortage of general and proved methodologies for dealing with knowledge processing. This means that systems can be produced that are understood only by the specialists that built them and formally

unverifiable. For this reason, a concentrated effort on developing methodologies for identifying, implementing and verifying knowledge based systems, is essencial.

- So far, there has been a lack of reasoning from first principles. This is inevitable, considering the immense body of knowledge that would have to be assembled to make up even a small portion of an engineer's "common sense." As a consequence, knowledge based systems cannot extend their knowledge to new fields; conversely, knowledge based systems don't know what they don't know, or when they go beyond the limit of applicability of their knowledge.
- The technology is new enough that it is still in the phase of a multitude of incompatible and immature competing tools, pending standardization (e.g., ANSI standard knowledge representations), and an industry shakeout. While the tools available can be of real value in appropriate cases, unless one is careful, systems can be developed that are not compatible with knowledge sources, and which have no migration path for upgrade or transformation to other representations.
- Typically the most sophisticated knowledge based techniques make inordinate demands on computing resources, requiring hardware such as LISP machine or multiprocessor networks. In a restricted hardware environment such as a spacecraft, it may only be possible to utilize a subset of the available techniques.

Anticipated Savings — The mission control system (MCS) of today consists of seven major task areas. The areas are flight planning, data load preparation, payload integration, training and simulation, flight control, communications and tracking, and post flight analysis. This and other studies point out many ways in which cost reductions can be achieved through the implementation of decision support systems, automated software production, and advanced information processing in the MCS areas.

Anticipated Risks — Expert system implementation introduces new risks which must be carefully managed. Some of the identified risks are:

- Problems may occur with verification and validation of the knowledge processing
- Automation might impact negatively on manual intervention, degrading safety
- Error propagation is more serious with KB systems which reduce data to conclusions
- Data security and system vulnerability problems
- Symbolic processing power required may exceed that which is available
- The problem may not be amenable to subdivision
- The cost of implementation may approach the level of anticipated savings
- The nature of the problem and knowledge required may change
- Minimal availability of top-level AI experts
- A possibly diffuse and poorly understood knowledge base

It should be emphasized that the verification and validation (V&V) of AI software is no simple task. Traditional software development methodologies use some form of path testing, measuring test coverage with metrics such a testing all branches of code. This testing often takes over 30% of the program effort. The concept of path testing does not mesh with AI systems well, because their static structure does not directly map onto their execution pattern as does traditional procedural code. For this reason the DoD software development standard is not yet applicable to the development and implementation of expert systems. It is perhaps obvious that an infinite number of tests is impractical, but the alternative is to release a program that may behave in ways not predicted by the designer. This is an issue that remains to be resolved. Our system integration laboratory (SIL) approach will be extremely useful in attacking the V&V problem, as imbedded systems become more intelligent. It is specifically aimed at hardware and software system integration.

The Abacus Expert System V&V Methodology is also a major risk reduction factor:

- Validate the inference engine separately from the knowledge base
- Utilize selcted expert system forms
- Maintain separate thoroughly validated tools
- Maintain a library of test cases and scenarios
- Use an independent panel of experts for system performance validation

AI TECHNIQUES

Knowledge-Based Representation— Although in certain cases performance or hardware compatibility may be the deciding factor, knowledge representation (KR) is generally the major issue determining the success of a knowledge based program. A successful KR must deal with the multifaceted knowledge that human experts bring to bear on problems, must be defined in terms of knowledge obtained from real sources, must be appropriate for the tools available, and must be defined with enough formality that it can be verified and validated without recourse to exhaustive testing. The knowledge engineer needs a toolbox containing a number of KR models (e.g., forward- and backward-chaining production systems, frames or schema, inheritance, escape to PROLOG, LISP, or procedural languages) which can be interlinked through knowledge gateways and/or blackboarding on standardized KRs. The Abacus AI Bus is a standardized knowledge gateway intended to coordinate distributed knowledge based processing systems.

Knowledge-Based Problem Solving — The techniques used for solving knowledge based problems cannot be separated from the KRs used to define the problems; solution techniques and representation techniques feed back and limit each other, and are both conditioned by the set of available tools. The basic issue is search, and the best techniques are those that reasonable narrow the search space most quickly, based upon intelligent use of knowledge rather than brute force. However, it is the problem KR that should drive the choice of techniques, subject only to a tool's ability to support them. What is important in a family of tools is the ability to link together a number of techniques where each technique is used where it is most appropriate. In situations where a system offering limited processing power is considered, a critical issue is the compatibility of the representations, for two reasons. First is the obvious requirement that the knowledge produced by one subsystem should be suitable for incorporation into another. Second, the behavior of the overall distributed system is difficult to verify if each element obeys a basic logic unrelated to the others.

Tool Evolution — Knowledge processing technology has advanced remarkably in the past decade, but is still relatively immature. In the next decade, tools will bring many of the current shortcomings up to a higher standard, as well as incorporating new features. The progress that is of most importance is not the latest research developments, however, but rather the standardization and certification of the already-existing technology so that it can be confidently utilized in more critical situations:

- Tools will aid in capturing and verifying knowledge:
  1) Built-in domain knowledge of selected application areas will require only add the specific parameters of the problem (i.e. electronic diagnostic systems already know how to interpret a circuit diagram).
  2) High-quality debugging tools to view a knowledge base in a useful graphic form and check it for completeness and consistency.
  3) Mixed representations allow portions of a problem to be expressed in different ways, for example a slot in a frame taking a value from the operation of a rule or from a database.
  4) Bridges to the outside world and data sources, such as databases, spreadsheets, and communications.
- Knowledge representation will be standardized and processors certified; analogous to the development of ANSI or DoD standards for programming languages, and the certification of compilers. In this way, knowledge bases conforming to a standard representation (i.e. forward-chaining production systems) will be portable from one tool to another in the way that FORTRAN is transported from one compiler to another.
- Routine use in real-time applications will come from improved hardware, tools that are optimized to certain hardware, and the progressive identification and elimination of bottlenecks as the commonalities in common knowledge based applications become more well known.

TECHNICAL APPROACH

The STRESS Program follows a progressive plan of development:

- **Task I** – Currently in progress, we have assessed key drivers to cost, schedule, safety, and mission success by filtering inputs from the STAS, MPRAS, and AGN&C studies; and are now deriving KBS candidates with a cursory analysis of V&V effects on these drivers.
- **Task II** – Develop a technology plan so as to identify the critical performance areas and propose a research schedule to support advanced vehicle devleopment.
- **Task III** – System partitioning predesign and tradeoffs will determine a maximum degree of autonomy that is performance and cost

effective. Evaluating the viable architectures as to benefits and risks will determine resultant effects to the onboard and ground systems requirements. Establish system specifications which will encompass the range of options identified in the predesign for each system and develop a feasibility demonstration plan that would be most representative of the critical functions and interfaces.

- **Task IV** — Develope pathfinder demonstration with functional integration to conventional avionics systems to show merit in an on-board environment.
- **Task V** — Prepare the documentation covering each of the above tasks, submit a final report for AF review, and present a final briefing of the study results.

STRESS Study Goals — The following goals are used as a point of departure:

- •Decreased costs achieved with:
  - -Increased autonomy, to minimize ground support time and personnel
  - -Improvements in the methods used today
  - -Minimized Mission Control Support by automation and vehicle autonomy
  - -Reduced post-flight analysis through on-board fault logging and testing
  - -Development of a standardized payload interface
- •Schedule Compression will result from:
  - -Availability allowing increased launch rates
  - -Deferred maintenance allowed by expert system redundancy management
  - -Autonomous vehicle approach, allowing automated checkout
- •Improved Mission Success will result from:
  - -Adaptive reconfigurable systems and control
  - -Improved depth of verification/checkout:
    1. Failure prediction by data trend analysis
    2. Improved data collection and correlation
- •Increased Flight Safety will result from:
  - -Human operator cross-check and backup
  - -Failure prediction allowed by trend analysis
  - -Hierarchical end-effect failure checking
  - -Replacement of men in hazardous operations

TOTAL SYSTEM DESIGN SOLUTIONS

On-Board/Ground Partitioning — The design goal is to determine the maximum degree of autonomy that could be delegated to the on-board Vehicle and Mission Management system, and establish the requirements for the complementary ground support system.

The structure of the VMMS will be based on the Pave Pillar architecture, figure 3 (without shaded interfaces), as modified by two complementary studies - the Autonomous Guidance, Navigation and Control (AGN&C) study and the Multipath Redundant Avionics Study (MPRAS). These study results, which are considering alternatives such as portioned vehicle autonomy, integrated smart sensors, and power/thermal/time optimization, will be incorporated into an overall system approach.

Maximum Degree of Autonomy — Studies to date have indicated that one of the most beneficial approaches which can be taken to reduce cost of launching vehicles is to strive for the maximum degree of autonomy for the on-board systems. The rational is obvious; the reduction of ground support personnel team required for a vehicle system saves money in a direct and understood manner. The airborne functions considered include:

- •Flight control reconfiguration
- •Fault tolerance management, including self diagnostics, fault - detection, identification and prediction
- •Man/Machine interfaces
- •Information gathering and management
- •Fuel management
- •Payload management
- •Mission planning, scheduling, replanning, and rescheduling
- •Computational resources management
- •Space/Space and Space/Ground telemetry data processing management
- •Performance management
- •Guidance and Flight control management

The "AI BUS": an Integrating Architecture — The AI Bus is an architecture developed by ABACUS to provide a highly flexible structure for integration of multiple expert systems and conventional systems hosted on a mix of machines in a distributed network. The AI Bus is not a physical bus, but instead a logical one that provides standard utilities and services through standard interfaces to both knowledge based and conventional software systems. This approach allows growth towards the increased use of expert systems by providing a common framework at initial design time, along with high technology transparency for attached components, This permits substitution of components, such as a nodal parallel processor, later in the design cycle than a more conventional architecture, and it mitigates the

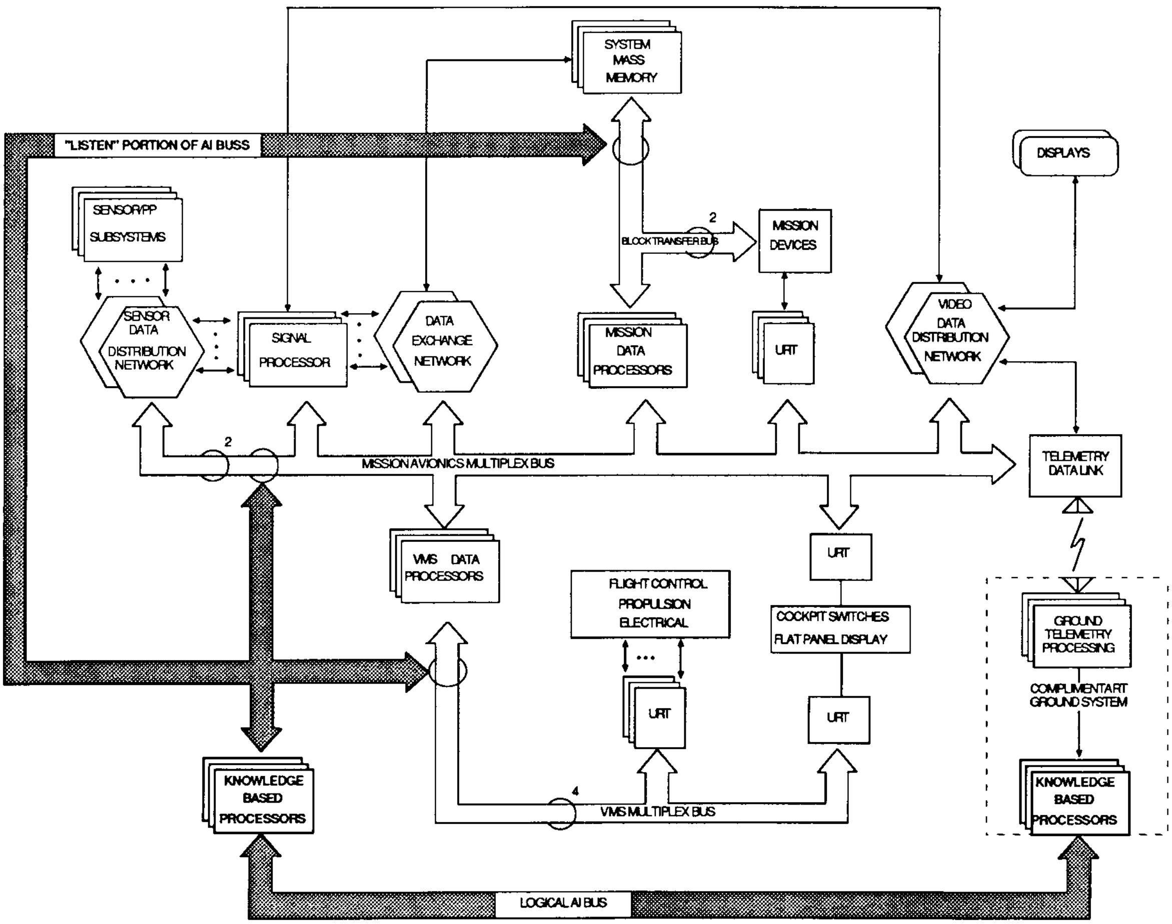

Figure 3. An AI Bus concept will provide inherent integration of knowledge based systems.

schedule risk issue. The bus is a layered architecture implemented upon the distributed data network of the vehicle avionics systems. The layers extend at the higher layers to the complementary ground based system.

Figure 3 (with shaded interfaces) shows the primary components of the AI Bus as it could be applied to the Pave Pillar architecture after it has be structured for the VMMS function. It includes:

- •Common code for inference engines
- •A knowledge based management system for handling working memory, truth maintenance, various knowledge representation techniques and standard database type access
- •Invocation scheduling based on system events and access to the event information
- •Access to system global information
- •A shared, distributed blackboard structures for knowledge based systems

The AI Bus differs from conventional knowledge based systems frameworks in that it:

- •Provides a "listen" capability for knowledge based systems to monitor other systems
- •Is designed to operate in, and take advantage of, a distributed architecture
- •Is designed to handle real time as well as conventional, consultive type knowledge based systems
- •Handles a hybrid combination of rule based and conventional procedural programs

CONCLUSIONS

Experience from our launch vehicle programs and other studies show that there are many opportunities in operations that reduce costs and improve autonomy, including:

- •Ground operations: daily planning support and timely work-around decisions aids
- •Ground checkout: autonomous procedural operations and control, standard trends, and red-line monitoring
- •On-board systems: monitoring, integration, and control
- •Launch day: fly with fault diagnostics and

decision aids
•Post-flight: data reduction and analysis

This program will assess these benefits vs implementation risks and demonstrate key performance requirements to show feasibility. Development of an overall integrated architecture will be important in providing a context and focus for the follow-on demonstration prototypes, and assure the synergy of their gains in both development and use in vehicle operations.

Phase II, beginning 3Q '88, will develop, demonstrate, and validate the cost reductions predicted by the use of expert decision aids in areas that would improve ground and on-board system autonomy. Proof of concept demonstrations will be selected in order to reduce the risk of commitment to this new technology. Each demo will be of a fractional scale; sufficient to give a good performance correlation to a full-scale implementation. These demonstrations will incrementally show:

- •Ease of human interface for maintainability
- •Real-time system performance
- •Integration to vehicle/ground hardware, and data systems
- •Validation methods consistent for ground and on-board applications.

Integration to the other related technology projects is essential to this approach. Figure 4 shows the flow of requirements, analyses, tool sets, standards, and interfaces between them. Final validation of these cost reductions will be done through demonstrations integrated into a "hot bench" environment to be established at NASA/KSC.

INTER-RELATED PROJECTS		EXPERT SYSTEMS FOR DECISION AIDS
INTEGRATED HEALTH MONITORING	←	REAL-TIME PERFORMANCE DATA TO SIZE DIAGNOSTIC & DATA COMPRESSION TECHNIQUES
IHM	→	DEMO IN OIL MULTIPLE COOPERATING HEALTH MANAGEMENT DECISION AIDS
MULTI-PATH REDUNDANT AVIONICS	←	REQUIREMENTS & ANALYSIS FOR DATA & KNOWLEDGE BUS TRAFFIC FOR ON-BOARD
MPRAS	→	INTERFACE TO ALS SYSTEMS INTEGRATION LAB (SIL) FOR INTEGRATED CHECKOUT
ADAPTIVE GUIDANCE, NAV, & CONTROL	←	STANDARDS FOR H/W, S/W, INTERFACE's, TOOL SETS, & VERIFICATION/VALIDATION REQUIREMENTS
AGN&C	→	REQUIREMENTS FOR FLIGHT CONTROL ANALYSIS & TRAJECTORY DESIGN TOOLS
ADVANCED MISSION OPERATIONS	←	STANDARDS FOR H/W, S/W, INTERFACE's, TOOL SETS, & VERIFICATION/VALIDATION REQUIREMENTS
	→	REQUIREMENTS FOR LABOR REDUCTIONS ON PLANNING, CHECKOUT, & DATA TRACKING
AUTOMATED GROUND PROCESSING	←	STANDARDS FOR H/W, S/W, INTERFACE's, TOOL SETS, & VERIFICATION/VALIDATION REQUIREMENTS
	→	REQUIREMENTS FOR DATA ANALYSIS, AUTOMATED CHECKOUT AND INTERFACES
OPERATIONS ENHANCEMENT CENTER	←	PROTOTYPE OPERATIONS, INTERFACE STANDARDS, INTELLIGENT CHECKOUT, AND TRAFFIC REQUIREMENTS
OEC	→	OPERABILITY, MAINTAINABILITY REQUIREMENTS
NASA/KSC	←	PROTOTYPE OPERATIONS, INTERFACE STANDARDS, INTELLIGENT CHECKOUT, AND TRAFFIC REQUIREMENTS
	→	PERFORMANCE DATA FOR AUTOMATED CHECKOUT AND INTERFACE CONSTRAINTS

Figure 4. The information flow between these interrelated technologies is essential to our approach.