AI Magazine Volume 7 Number 3 (1986) (© AAAI)

Kathleen Jurica Healey, Editor

Artificial Intelligence Research and Applications at the NASA Johnson Space Center

Mission Planning and Analysis Division Technology Development and Applications Branch

The Technology Development and Applications Branch of the Space Operations Directorate has the following functional responsibilities: (1) the evaluation, assessment, and implementation of new technology from NASA research centers, industries and universities relative to JSC space operation applications; (2) development of in-house capabilities for assessment and implementation of new technologies; and (3) prototype development and demonstrations of new technologies. Specific applications of artificial intelligence include robotic simulations, interactive graphics, information communication, man-machine interface, expert systems, automated software development, intelligent computer-aided instruction, intelligent computer-aided engineering, expert system builders, parallel processing, and software verification and validation.

Robotic Systems Development

This project will develop a robotic software test bed that integrates artificial intelligence and graphics system technologies for dynamic system simulations. Specific objectives are (1) to develop a clearer understanding of the computing, communications, and control capabilities needed for coordinated robotic manipulation of objects in a space-based environment; (2) to develop an understanding of trade-offs and the complementary relationships between graphic simulation modules and empirical test-bed modules with respect to robotic subsystem design, development, and proof-of-concept demonstrations; (3) to identify requirements for integrating coordinated intelligent robotic units with the evolutionary space station network of computing systems, communications and control systems, and common service and hardware-software elements; and (4) to identify short- and long-term technology needs. In the process, technical issues such as perception, automated aids for problem solving, cognition and learning, adaptive fault tolerance, and safety will be evaluated for applicability to the use of manipulators in space.

Participant: C. Gott, JSC (contact)

Distributed Expert Database System

MPAD is developing a distributed database management system that integrates all centralized database systems and software developed by its various branches. Major objectives are to eliminate data redundancy and to provide access to all MPAD centralized databases and software by all MPAD users, regardless of user and database and software site locations. Cooperative knowledge base systems will also be developed to provide a highly sophisticated user interface to the distributed database management system for problem solving.

Participant: E. Fridge, JSC (contact)

Abstract

This is the second part of a two-part article describing AI work at the NASA Johnson Space Center (JSC).

Research and applications work in AI is being conducted by several groups at JSC. These are primarily independent groups that interact with each other on an informal basis. In the Research and Engineering Directorate, these groups include (1) the Artificial Intelligence and Information Sciences Office, (2) the Simulation and Avionics Integration Division, (3) the Avionics Systems Division, and (4) the Tracking and Communications Division. In the Space Operations Directorate, these groups include (1) the Mission Planning and Analysis Division (MPAD)—Technology Development and Applications Branch, (2) the Spacecraft Software Division, and (3) the Systems Division—Systems Support Section. This second part of the article describes the AI work in the Space Operations Directorate The first part of the article, published in the last issue of *AI Magazine*, (7:1, Summer 1986) described the AI work in the Research and Engineering Directorate

Kathleen Healey is Special Assistant at the Simulation and Avionics Integration Division, Research and Engineering Directorate, NASA Johnson Space Center, Houston, Texas 77058.

Software Development User Interface

An artificially intelligent graphics system is being developed to replace the present tedious and error-prone computer programming language techniques for interfacing and ordering software subroutines. This system will be evaluated to determine its enhancement of the productivity of software engineers.

Participant: B. Hinson, JSC (contact)

Software Architecture Prototyping

The goal of this project is to provide an AI system that can be used by a general user (one who does not have programming knowledge) to create and verify software application systems. The key elements of the AI system are a library of existing software routines, heuristics for merging software, and an inference engine that accesses the library and rule base to build software application systems based on user input. Two steps are involved in the process. The first step merges routines to form application elements. The second step sequences these application elements to produce the final software application system.

Participant: E. Fridge, JSC (contact)

AI Section

The AI Section supports multiple automation objectives: early demonstrations of automation applications, design of software development environments to be compatible with and supportive of AI, the minimization of lead time from demonstration of technology to operations, development of knowledge-based automatic programming systems, and the use of existing technologies from industry and other sources to the maximum extent possible.

The artificial intelligence laboratory hardware includes nine Lisp computers and one VAX computer. The Lisp computers are Symbolics (2-3670, 5-3640), LMI, and HP (1-9836).

The AI Section evaluates AI hardware (including compact Lisp machines) and AI software (Zetalisp, Common Lisp, PROLOG, SMP, MACSYMA, ART, KEE, OPS5, MRS, microcomputer expert system building tools, cross-compilers [Lisp, C, ADA, and so on]) for ground and on-board applications. The AI Section is also involved in the development of expert system applications and expert system builders and AI training.

Current activities focus on several areas of interest: real-time monitoring and control, process planning and control, automated software development and synthesis, intelligent interface development, AI software development, and parallel processing.

Contacts for all activities are Robert Savely or Sandy Richardson, JSC.

Navigation Expert (NAVEX). This is an expert system that evaluates the decision-making process of the flight controllers who work on the high-speed navigation console during the ascent and entry phases of shuttle missions. The console task currently requires three controllers who monitor the tracking data from 1, 2, or 3 radar stations, control the operation of the high-speed trajectory determinator (HSTD), monitor the comparison between the on-board and ground navigation systems, and provide status information to the flight dynamics officer. The expert system monitors the same radar data and output from the HSTD. It warns the operator of current or impending problems with the data and recommends potential actions. This system has the capability to reduce manpower requirements from three people each shift to one.

Participants: Chris Culbert, JSC and Mark Maletz, Inference Corporation

References: Maletz & Culbert (1985), Culbert (1985)

On-Orbit Navigation. This is an expert system used to consolidate a number of functions that analyze and monitor shuttle data during flights. It uses high-speed data lines from the Mission Control Center (MCC) into an HP9000. The data processing occurs in both real-time and near–real-time modes. The orbit determination portion of this project (Kalman filter) is complete. The expert system was developed on the Symbolics to monitor and control the Kalman filter. The system automates data collection and preparation, analyzes the filter output, and controls processing. It also makes recommendations on state-vector quality and provides an automatic check on filter divergence, performing an automatic restart when divergence is detected. This expert system has been successfully tested in near–real-time mode for missions STS 51-F and STS 51-I using tactical air control and navigation (TACAN) detailed test-objective data.

Participants: Paul Mitchell and Gary Riley, JSC

MCC Software Status Expert System (MCCSSES). This is an expert system that emulates the function of the printer controller in the MCC. This printer controller monitors the online printer for error status messages during all flights and simulations. Virtually all the software and hardware in MCC report error or status information to this printer. The printer controller scans this printout in real time for significant information and then reports it to other flight controllers, primarily the computer supervisor. The MCCSSES could potentially eliminate this job and could also provide extended capability for error detection, analysis, and correction.

Participants: Penny Clemons, Lui Wang, Patty Herrick, and Chris Culbert, JSC

References: Clemons, Culbert & Wang (1985), Culbert (1985)

Applications of Nondeterministic Methods (Fuzzy Logic) to Space Autonomous Problems. *Fuzzy logic* is an attempt to apply commonsense reasoning to problems where rules and data are not precisely known. This test project used star-tracker shuttle data because definite problems can arise in processing this data if the crew does not monitor incoming data. An algorithm based on fuzzy sets was developed to generate a decision for disposing of shuttle navigation sensor measurements during rendezvous operations. The algorithm is used in a prototype system to monitor sensor measurements and to make the decision to process or not process. Fuzzy-set characteristic functions are used to generate the decision and to classify the measurements according to their degree of belief, that is, the value of a fuzzy-set degree-of-membership function prior to using the measurements to update the relative state. The algorithm has been tested against the performance of the crew on actual flight data, including nominal and nonnominal data. The sensitivity of the decision function to imprecisely defined parameters has also been examined. The results indicate that the algorithm performs as well as the current Kalman filter on nominal data and that it performs better than the current filter on problem data without crew assistance.

Participant: Robert Lea, JSC

On-board Navigation Expert System (ONEX). This is an expert system that automates real-time monitoring of on-board hardware and software statuses affecting onboard navigation for rendezvous. Prototypes were developed for the MCC upgrade and for space station development. The expert system currently obtains near–real-time downlinked data through a local-area network (LAN) from MCC to a workstation. It operates on a set of monitoring rules for nominal rendezvous.

Participants: Paul Mitchell and Mary Ann Goodwin, JSC and Benita Talisman and Glenn Goodrum, McDonnell Douglas

ES Shuttle Electric Power Management (EXEPS). EXEPS aids in scheduling spacecraft electric power system activity blocks. Activity blocks are aggregates of electric components. JSC characterizes the expected performance of the electric power system before each flight based on that flight's schedule of trajectory events and crew activities. This task requires expert knowledge of the shuttle's electric power system and currently takes from four days to two weeks. EXEPS will function as an intelligent adviser, helping the user obtain an activity block time line that both accurately reflects the known flight events and satisfies a set of predetermined constraints. The EXEPS prototype, built in Zetalisp and ART (from Inference Corporation), allows users to interact with the activity block schedule by way of a graphic time line of activity blocks and crew events. The ART program schedules activity blocks that are then displayed in the object-oriented interface developed in flavors. Each activity block and crew event is both a flavor instance and some number of ART facts, allowing the program to use both object-oriented and rule-based reasoning. The integration of these two tools was the major technical innovation of the EXEPS project.

Participants: Marlon Boarnet and Dave Heath, JSC

References: Heath & Boarnet (1985), Culbert (1985)

Intelligent Interface to the Mission Requirements Database (MRDB). The space station MRDB contains information on 326 proposed space station missions. A mission is any customer payload or experiment and, as such, can require many pieces of equipment for possibly long periods of time. MRDB contains resource usage information on each mission. The AI Section examined building a friendly graphic intelligent interface to the MRDB using the object-oriented programming facilities of KEE (from IntelliCorp) with Zetalisp code for procedural programming. The intelligent interface allows users to alter the schedule of missions by moving icons on a time line. The intelligent interface displays histograms of resource usage that always reflect the latest schedule of missions. This process provides a modeling capability that allows users to examine different scheduling alternatives. The intelligent interface also flags constraint violations as they occur and offers limited resource-leveling capabilities. Future versions of the intelligent interface could offer suggestions about resolving constraint violations and could automatically schedule space station missions. A fast production system will be used for this reasoning portion. Future development might also include scheduling daily crew activities and performing detailed premission analysis as MRDB information becomes more detailed. No commitment has yet been made to expand this tool to include automatic-scheduling capabilities.

Participants: Marlon Boarnet and Dave Heath, JSC

Resource Planning and Management System (RPMS). This is a scheduling shell that uses flavors (for the graphic user interface) and an OPS5 inference engine. It allows users to schedule events, define precedence relations between events, and attach resources to events. All manipulation of events and resources is done graphically, using a mouse. Users can dynamically alter the schedule, and as it is altered, the program preserves all unchanged precedence and resource information. Users can also define constraints that will trigger warnings when violated. This system inspired later work on scheduling programs, such as EXEPS and the space station MRDB. It is under evaluation both as a basic model for scheduling programs and as a manifesting tool.

Participants: Brooks Pollack, Ford Aerospace and Marlon Boarnet, JSC

Ground Navigation Console Support Scheduling Expert System. This expert system schedules console support for simulations and missions. It considers mission constraints, experience levels, and personal preferences in producing the schedules. The system allows interactive scheduling with what-if situations. It was prototyped on a Symbolics machine, ported to the IBM PC using Golden Common Lisp, and is currently being tested by users.

Participants: Gary Riley and Lui Wang, JSC

Expert System for the Flight Analysis System (ESFAS)

This is an expert system that acts as an intelligent interface to the flight analysis system (FAS), which is a set of computer programs used to design many of the shuttle missions. The system contains knowledge about the FAS processors, the executive, and methods of applying processors to problems. The objectives of this system were to design a structure that would support incremental development and to automate knowledge about the FAS. Other objectives were to demonstrate the potential of AI tools to build user-friendly interfaces and to create a mission trajectory event time line for any number of orbital adjust maneuvers. The product consists of a sequence table file and all necessary interface table files to execute FORTRAN programs.

Participants: Lui Wang, Chris Culbert, and Paul Mitchell, JSC and Harvey Flinn, Barrios Technology

References: Culbert, Wang and Flinn (1985), Culbert (1985)

Intelligent Workstation Interface. This project entails an interface to the user which is intelligent in the sense that it supports natural input and guidance for the user. The AI Section is investigating various methods of communicating with computer systems. These include voice recognition and synthesis, various natural language grammars and parsing techniques, and computer graphics. The project currently under way is that of applying voice recognition and/or synthesis and natural language to ESFAS. There are basically three parts to this intelligent interface: (1) Language Craft: A natural language parser from Carnegie Group, Inc.; (2) Dectalk: A voice synthesizer; and (3) Kurzweil Voice System: A speech-understanding system from Kurzweil A.I.

The lessons learned from this project will then be applied to the development of command and control workstation test beds.

Participants: James Villarreal, Sandy Richardson, Lori Wright, Monica Rua, Lui Wang, and Chris Culbert, JSC

Magnetoencephalography for Real-Time Computer Control (Small Business Innovation Research). The principal objective of this research is to investigate and verify the concept of sensing brain waves for computer control. The ultimate benefits involve the control of computers through brain waves. Other benefits could include advances into the understanding of neurological disorders. This objective will be met by examining the following issues:

Brainwave-sensing technology assessment: This subtask will examine the current state of the art in sensing hardware, techniques, and characteristics. Traditional brain-wave sensing involves the interfacing of the neural system with electrodes attached to the scalp (electroencephalography [EEG]). The proposed research is innovative in that the technique of interest is interfacing the neural system by detecting magnetic fields emanating from the brain (magnetoencephalography [MEG]).

Overall system assessment: This subtask will examine the proposed concept from a total system perspective.

Final report and Phase II plan: This subtask will document Phase I efforts and results. Based on the results from subtasks 1 and 2, a Phase II development plan, which includes a feasibility demonstration, will be developed.

Participants: J. Villarreal and M. Boarnet, JSC

C Language Intelligent Production System (CLIPS). This system was developed by the AI Section as a forward-chaining rule-based language. It is portable, written completely, and fully integrated with C. It is used to help define expert system builder requirements. Plans are to use CLIPS as a training tool and as a prototype for other expert system builders.

Participants: Gary Riley and Frank Lopez, JSC

Next-Generation Expert System Tool Requirement Development. The AI Section is conducting an ongoing examination of the necessary capabilities of next-generation expert system building tools. This study considered three major AI software products—ART, KEE, and Knowledge Craft—to define the state of the art in expert system tools. Using these product findings as a departure point, the study attempted to pinpoint improvements necessary for NASA expert system needs. All improvements are projected to be technically feasible in the next one to four years (from August 1986) with most projected to be feasible in one to four years.

Although the study of future expert system builders is still undergoing internal NASA evaluation, many needed improvements have already been noted. Future tools must give more attention to user interfaces, making interface development a natural part of expert system development and providing adequate representational tools for interface development. Different representations offered in hybrid tools (most often some combination of rules, frames, and objects) must become more integrated and more flexible, thereby allowing a user to use any or all of the representations and to allow rule systems to reason about the information in the frame or object network. Future tools should provide architectural support for distributed and parallel processing. Both the hardware and software environments must become more standardized to allow greater code portability and more delivery alternatives.

Within MPAD, it is generally believed that graphic knowledge base editors have great value both as teaching tools and as on-screen knowledge base organizers. Still, the readability of text editors is also valuable; hence, more development appears needed to combine the advantages of both approaches.

Participants: Marlon Boarnet, JSC and N. S. Rajaram, University of Houston

References: Rajaram (1985), Rajaram & Tsehaie (1985), Rajaram (1986)

Expert System Construction Tools in Ada. The goal of this project is to develop Ada software tools to aid in the construction of Ada-based expert systems. (Traditionally, expert systems execute in a Lisp environment, which is considered by MPAD to be unacceptable for long-term operations because Lisp environments are optimized for development and not operations.) The Ada tools will provide a production system environment for rule-based expert system development and execution. Ada facilitates the primary objective of providing portable tools on "standard" minicomputers as opposed to dedicated Lisp machines. The tools should incorporate a number of capabilities to allow versatile development of robust expert systems, including inferenceing, higher-order language definition, data- and control-structure definition, and manipulation capabilities. Hypothetical reasoning, forward- and backward-chaining rule structures, object-oriented programming, the capability to call routines in other languages, and simple debugging aids will also be included. The development effort will be split into three phases: (1) initial design specifications and development of a prototype system; (2) complete design specification, expansion of the prototype system, and initial testing; and (3) final prototype, full verification and testing of capabilities, documentation of the system, and implementation on MPAD workstations. Each phase will have design status reviews and demonstrations of the current prototype.

C-Based Expert System Shell for Real-Time Applications (Small Business Innovation Research). The objective of this project is to develop a tool for building portable event-driven expert systems for real-time applications. The tool will support multiple, cooperating expert systems; real-time response to events; and asynchronous event handling. (Multiple, cooperating expert systems are potentially very important for autonomous on-board control of the space station and other space-based systems as well as commercial process monitoring and system control.) The architecture of the event-driven expert systems is to be based on the HEARSAY "blackboard" structure. An extension of the blackboard structure will allow interrupt processing for high-priority events. The rule structure and inference engines will be based on OPS5+, developed by Artelligence.

Participant: Chris Culbert, JSC

Parallel Processing. Additional performance increases are expected to be achieved with parallel processing. The AI Section has obtained a Flex parallel computer with which it will investigate the use of parallel processing in AI applications. The Flex can house from 1 to 20 processors in a single cabinet, and several cabinets can be linked together to provide a much larger number of processors. The parallel languages supported by the Flex include Concurrent C, Concurrent FORTRAN, and Ada. Concurrent C and Concurrent FORTRAN contain standard C and FORTRAN 77, respectively, with language extensions for concurrent processing. The concurrent language extensions provide a natural and easy-to-use method for incorporating concurrent processing into C and FORTRAN programs. The AI Section has ported its expert system shell, CLIPS, to the Flex on which it currently executes in a nonconcurrent UNIX environment. Future plans include converting CLIPS to run in a concurrent environment as well as investigating the employment of AI techniques in general in a concurrent environment.

The concurrent environment will exploit the parallelism inherent in the RETE pattern network that is typically used to determine the conflict set in production systems. Each condition element in the left-hand side of a rule becomes a top-level node in the network, and lower nodes enforce variable bindings. Terminal nodes represent successfully matched (instantiated) rules. The set of all instantiated rules forms the conflict set. One possible scheme would allow one pattern for each processor, with pattern-net binding done in a single massively parallel operation. Join-net computation would be done by interprocessor communication and conflict resolution, and firing (acting) would still be done sequentially. Another scheme, one that is the focus of much discussion within the AI Section, would place one rule at each processor, allowing rules to be handled in parallel. The matching would be done sequentially for each rule, but all rules could be matched simultaneously. With this scheme, one could either retain the usual sequential conflict resolution and fire schemes, or one could add greater parallelism. With one rule to a processor, one could skip conflict resolution and fire any rule as soon as it matches facts in working memory.

(This would require working memory to be in a completely shared area of memory.) This would change one of the fundamental operating characteristics of production systems. However, the change would be acceptable if the rules are truly independent pieces of knowledge that can be fired at any time after they have been instantiated.

Participants: Gary Riley and Chris Culbert, JSC

Spacecraft Software Division

The primary responsibilities of the Spacecraft Software Division include development and maintenance of the on-board shuttle software and reconfiguration tools. For the space station, the division will provide a standard set of tools and facilities for software development, provide the on-board system environment, and play a key role in the development of the user interface language. This leads to four goals with respect to artificial intelligence: (1) ensuring that the space station software-support environment contains tools for the effective development of AI software, (2) providing a framework in the on-board data management system (DMS) for the delivery and integration of AI systems, (3) utilizing AI techniques to improve the operation of DMS and the user interface language, and (4) utilizing AI systems to facilitate the integration and configuration of large software systems.

In order to meet these goals, the division monitors trends and methodologies used in the development of AI systems and engages in selected prototype projects to further explore and gain experience in specific areas. Two prototypes are currently under development.

Space Shuttle Computer Diagnostic Expert System Task

This task provides a prototype expert system that diagnoses space shuttle computer failures. The Phase 1 proof-of-concept work has been completed successfully. It was developed on the IBM VM/370 system located at the IBM Federal Systems Division facilities near JSC. The product used by IBM to develop the system is the now commercially available Expert System Environment/VM. Two important characteristics of this system are (1) that it runs on readily available equipment using the standard VM operating system, and (2) that it interfaces directly with shuttle computer memory dump data and, hence, is an example of an expert system providing results based on interacting with real-world data.

The purposes of this task are (1) to demonstrate that expert system technology is sufficiently mature to allow the rapid prototyping of a cost-effective diagnostic capability for a complex domain like a space-based computer system, (2) to demonstrate the capabilities of currently available tools for building these systems, (3) to show the capabilities of local personnel (not AI researchers) in using these tools and producing such a system, (4) to develop a pragmatic set of knowledge engineering methods for solving real-world problems by attempting to solve such a problem, and (5) to provide an effective tool for diagnosing shuttle computer failures.

Future goals for the task are (1) to refine the current system to cover a more complete set of failure conditions, including software problems; (2) to extend the system to cover the next generation shuttle computer, the AP101S; and (3) to bring the system to an operational level of capability.

Participants: S. Gorman, JSC (contact) and L. Pennington, IBM

Data Management System (DMS) Intelligent Executive Prototype

The goal of this task is to illustrate AI concepts for man-machine interaction and subsystem management. The first phase concentrates on a natural language interface with the following objectives: (1) To support voice input and output; (2) To support flexible, adaptive interaction with users; and (3) To support development of methods for modeling and manipulating knowledge about system components and procedures.

Later phases will add online "expert" help facilities; automatic monitoring and control of DMS subsystems; and, ultimately, an integration point for "reasoning" with, and acting on, the conclusions and recommendations of subsystem AI elements that might affect other subsystems.

Currently, the project is in its first phase, and alternate methods of implementing natural language interfaces are under investigation. The interface selected must be suitably general and expandable.

Participants: R. Shuler (contact) and C. Foster, JSC

Systems Division—Systems Support Section

A primary task of the Space Operations Directorate at JSC is to provide ground support and crew training for the space transportation system. As the orbiter flight rate increases and the program moves toward an operational technical-scientific national resource, the challenge to provide quality products, support, and crew training seems insurmountable with existing approaches and facilities. Several efforts are under way to alleviate the work load being imposed on the present highly trained staff by applying automation techniques, especially in the telemetry monitoring and analysis area.

JSC Space Shuttle Fuel Cell Monitoring Automation Prototype

This project is being conducted to assess expert system, natural language development technology for use by the space shuttle mission operations community. All of the operations procedures are fully documented and verified (design, simulation, and mission data), and the ground flight controllers (domain experts) fully understand the decision-making reasoning. We believe the flight controllers can build their own expert system applications without the use of knowledge engineers if they are given a computer with an easy-to-use natural language interface. The orbiter fuel cells were chosen for the initial prototype because lack of sufficient instrumentation hampers isolation of the cause(s) of system degradation and failure.

Project design has been completed and initial prototyping, using KEE, is complete. Guidelines and instructions for use within the organization will be written so that expert systems for other vehicle subsystems can be developed, and the fuel cell prototype will be expanded to include the electric bus distribution subsystem. The individual expert systems will then be integrated to test interaction capabilities. Other objectives of the project are to test the expert systems developed by the flight controllers as subsystem familiarization and simulation tools for training new flight controllers.

Participants: Marilyn Kimball, JSC (contact) and Larry Minter, RSOC

Crew Interface Software —System Operations Development Laboratory (SODL)

This project focuses on the need for implementable operation requirements to levy on the Space Station Program during the early design phase. Automation of on-board (as well as ground) functions and a significant reduction in long-term recurring operational costs are two key factors in achieving the space station design goal. Ongoing developments in automation and modernization of the present MCC at JSC will be integrated with parallel automation and expert system prototypes into SODL to evaluate crew interfaces in the following technical areas: (1) on-board planning; (2) fault detection, annunciation, and reconfiguration; (3) on-board subsystem control; (4) displays and controls; (5) on-board training; (6) operations management system and database; and (7) in-flight maintenance.

A closed-loop prototyping evaluation environment will be maintained, using open LAN communications, with all major JSC (ultimately NASA) areas performing complimentary or similar research.

Some of the key areas to be investigated will be (1) distributed expert systems; (2) embedded system performance and analysis models; (3) total configuration and reconfiguration tracking; (4) total autonomous system management; (5) "smart" instrumentation; (6) streamlined crew interfaces, from keyboard to voice annunciation and recognition; and (7) automated crew procedures, training, and activity planning.

Additional areas requiring definition are growth, "hooks" and "scars," modularity, and accommodation (noninterference of system operation with science and commercial objectives).

Participants: William Bates (contact)and Dennis Webb, JSC

References

Clemons, P., Culbert, C. & Wang, L. 1985 "Development of an Expert System to Assist Monitoring Mission Control Center Software Status." In *ISA Workshop on Robotics and Expert Systems* 1: 113-115

Culbert, C 1985 The Role of Artificial Intelligence and Expert "Systems in Increasing STS Operations Productivity" In *R& D Productivity, 1985*, 378-382.

Culbert, C.; Wang, L.; & Flinn, H. 1985. "ESFAS: An Intelligent User Interface for the Flight Analysis System." In *ISA Workshop on Robotics and Expert Systems* 1: 215-220.

Heath, D., & Boarnet, M 1985. "EXEPS: An Advisory Expert System for Timelining Electrical Power System Activity Blocks " In *ISA Workshop on Robotics and Expert Systems* 1: 63-71.

Maletz, M , & Culbert, C. 1985. "Monitoring Real-Time Processes Using the Automated Reasoning Tool (ART) " In *Aerospace Applications of Artificial Intelligence 1985*, 350-360.

Rajaram, N S 1986 "Artificial Intelligence II—Expert Systems, the Cutting Edge " *Robotics Age* (forthcoming)

Rajaram, N. S. 1985. "Architectural Classification for Expert System Design." In *Conference on Artificial Intelligence and Advanced Computer Technology 1985*, 225-229.

Rajaram, N. S , & Tsehaie, H 1985 "Role of Expert Systems in Disaster Prevention " In *ISA 1985*, 1472-1475