N86-24525

1985

NASA/ASEE SUMMER FACULTY RESEARCH FELLOWSHIP PROGRAM

MARSHALL SPACE FLIGHT CENTER
THE UNIVERSITY OF ALABAMA

A STUDY OF THE VERY HIGH ORDER NATURAL USER LANGUAGE (WITH AI CAPABILITIES) FOR THE NASA SPACE STATION COMMON MODULE

Prepared By:	Esther N. Gill, Ed.D.
Academic Rank:	Associate Professor
University and Department:	Oakwood College Business and Information Systems Management
NASA/MSFC:	
Division:	Software & Data Management
Branch:	Systems Software
MSFC Counterparts:	John W. Wolfsberger and Robert L. Stevens
Date:	August 2, 1985
Contract No.:	NGT 01-008-021 The University of Alabama in Huntsville

A STUDY OF THE VERY HIGH ORDER NATURAL USER LANGUAGE (WITH AI CAPABILITIES) FOR THE NASA SPACE STATION COMMON MODULE

BY

Esther N. Gill, Ed.D.
Associate Professor of Computer Science
Oakwood College
Huntsville, Alabama 35896

ABSTRACT

This study will identify the requirements for a very high order natural language to be used by crew members on board the Space Station and will be a part of each module's common computer core operating system. The study will take into consideration the hardware facilities, databases, real-time processes, software support, etc. available or selected and will evaluate the operations and capabilities that will be required in both normal (routine) and abnormal (nonroutine) situations. The study will recommend a structure and syntax for an interface (front-end) language to satisfy the above requirements.

ACKNOWLEDGEMENTS

This researcher would like to acknowledge the contributions of her counterparts, Mr. John W. Wolfsberger and Mr. R. L. Stevens, whose aide was invaluable in obtaining source documents and other resources on the subject explored. Mr. Wolfsberger's enthusiasm for his work also motivated and stimulated this effort. I would also like to thank Mr. Dave Aichele, Mr. William Bradford, Mr. Jack Lucas, and Mr. Gabe Wallace for the privilege of entrance afforded me to the Software Evaluation Lab which made this research project possible. Thank you Mr. Jimmy Watkins for making my working surroundings more comfortable and for proofreading this paper.

NASA/ASEE Program Chairpersons, Mr. Thomas L. Osborn, Dr. James Dozier, and Dr. Gerald R. Karr, also are commended for their time and effort expended coordinating and administering the rewarding research experiences of the summer fellows. Thank you also Mrs. Dina Conrad for your personal helpful flare.

To my research colleague, Dr. Arthur Knoebel, gracias for making my first experience as a NASA/ASEE fellow a shared office pleasure.

I have gained a wealth of knowledge which will be taken back to my classroom and used for enriching the experiences of my students, for future proposal presentations to NASA for possible approval and funding, and for future NASA-employment, college-recruitment efforts.

Dr. Esther Naomi Gill

A STUDY OF THE VERY HIGH ORDER NATURAL USER LANGUAGE
(WITH AI CAPABILITIES)
FOR THE NASA SPACE STATION COMMON MODULE

T A B L E O F C O N T E N T S

GLOSSARY OF TERMS USED

AI HANDBOOK: The Handbook of Artificial Intelligence, E.A. Feigenbaum, A. Barr and P.R. Cohen (Eds.). Published by W. Kaufmann, Los Altos, CA in 1981 and 1982.

ARTIFICIAL INTELLIGENCE (AI) APPROACH: An approach that has its emphasis on symbolic processes for representing and manipulating knowledge in a problem solving mode.

BACKWARD CHAINING: A form of reasoning starting with a goal and recursively chaining backwards to its antecedent goals or states by applying applicable operators until an appropriate earlier state is reached or the system backtracks. This is a form of depth-first search. When the application of operators changes a single goal or state into multiple goals or states, the approach is referred to as problem reduction.

COMMON SENSE: The ability to act appropriately in everyday situations based on one's lifetime accumulation of experiential knowledge.

COMMON SENSE REASONING: Low-level reasoning based on a wealth of experience.

COMPUTER GRAPHICS: Visual representations generated by a computer (usually observed on a monitoring screen).

COMPUTER VISION (COMPUTATIONAL OR MACHINE VISION): Perception by a computer, based on visual sernsory input, in which a symbolic description is developed of a scene depicted in an image. It is often a knowledge-based expectation-guided process that uses models to interpret sensory data. Used somewhat synonymously with image understanding and scene analysis.

CONCEPTUAL DEPENDENCY: An approach to natural language understanding in which sentences are translated into basic concepts expressed as a small set of semantic primatives.

DATA BASE: An organized collection of data about some subject.

DATA BASE MANAGEMENT SYSTEM: A computer system for the storage and retrieval of information about some domain.

DATA STRUCTURE: The form in which data is stored in a computer.

DOMAIN: The problem area of interest.

EXPERT SYSTEM: A computer program that uses knowledge and reasoning techniques to solve problems normally requiring the abilities of human experts. Can capture knowledge of experienced engineers before they leave a firm, reclaiming years of "learning by doing" that can then be passed on to new engineers through the system.

FIFTH GENERATION COMPUTER: A non-Von Neumann, intelligent, parallel processing form of computer now being pursued by Japan.

FORWARD CHAINING: Event-driven or data-driven reasoning.

HEURISTICS: Rules of thumb or empirical knowledge used to help guide a problem solution.

HIGHER ORDER LANGUAGE (HOL): A computer language (such as FORTRAN or LISP) requiring fewer steatements than machine language and usually substantially easier to use and read.

IMAGE UNDERSTANDING (IU): Visual perception by a computer employing geometric modeling and the AI techniques of knowledge representation and cognitive processing to develop scene interpretations from image data. IU has dealt extensively with three-dimensional objects.

INFERENCE: The process of reaching a conclusion based on an initial set of propositions, the truths of which are known or assumed.

INFERENCE ENGINE: Another name given to the control structure of an AI problem solver in which the control is separate from the knowledge.

INTELLIGENT ASSISTANT: An AI computer program (usually an expert system) that aids a person in the performance of a task.

INTERACTIVE ENVIRONMENTS: A computational system in which the user interacts (dialogues) with the system (in real time) during the process of developing or running a computer program.

INTERFACE: The system by which the user interacts with the computer. In general, the junction between two components.

KNOWLEDGE BASE: AI data bases that are not merely files of uniform content, but are collections of facts, inferences, and procedures, corresponding to the types of information needed for problem solution.

KNOWLEDGE ENGINEERING: The AI approach focusing on the use of knowledge (e.g., as in expert systems) to solve problems.

NATURAL LANGUAGE INFERENCE (NLI): A system for communicating with a computer by using a natural language.

NATURAL LANGUAGE PROCESSING (NLP): Processing of natural language (e.g., English) by a computer to facilitate communication with the computer or for other purposes such as language translation.

NATURAL LANGAUGE UNDERSTANDING (NLU): Response by a computer based on the meaning of a natural language input.

PERSONAL AI COMPUTER: New, small, interactive, stand-alone computers for use by an AI researcher in developing AI programs. Usually specifically designed to run an AI language such as LISP.

PROGRAMMING IN LOGIC (PROLOG): A logic-oriented AI language developed in France and popular in Europe and Japan.

SEMANTIC: Of or relating to meaning.

SOFTWARE: A computer program.

SPEECH RECOGNITION: Recognition by a computer (primarily by pattern matching) of spoken words or sentences.

SPEECH SYNTHESIS: Developing spoken speech from text or other representations.

SPEECH UNDERSTANDING: Speech perception by a computer.

SYMBOLICS: Relating to the substitution of abstract representations (symbols) for concrete objects.

SYNTAX: The order or arrangement (e,.g., the grammar of a language).

VON NEUMANN ARCHITECTURE: The current standard computer architecture that uses sequential processing.

LANGUAGES

CATEGORIES	CANDIDATES?
Requirements and specification	PSL/PSA, SREM, SADT, CADSAT
Design	PDL, SDDL
Development	HAL/S, Fortran, PL/I, Jovial, Ada, C, Modula-2, Pascal
User interface	GOAL, ATLAS, SCOL,STOL, Ada
AI/expert systems	LISP, PROLOG

ILLUSTRATION 2

PROGRAM SUCCESS DEPENDS ON SOFTWARE

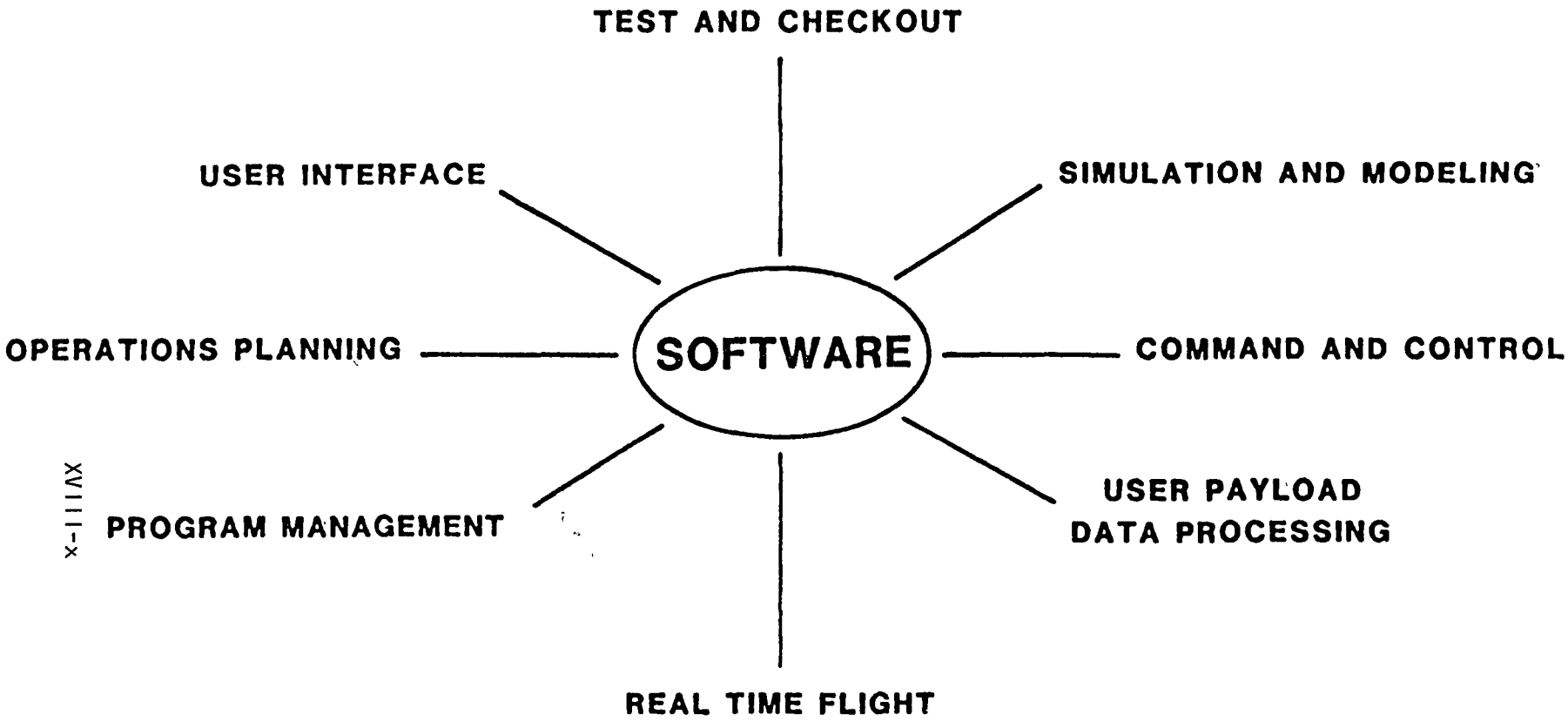
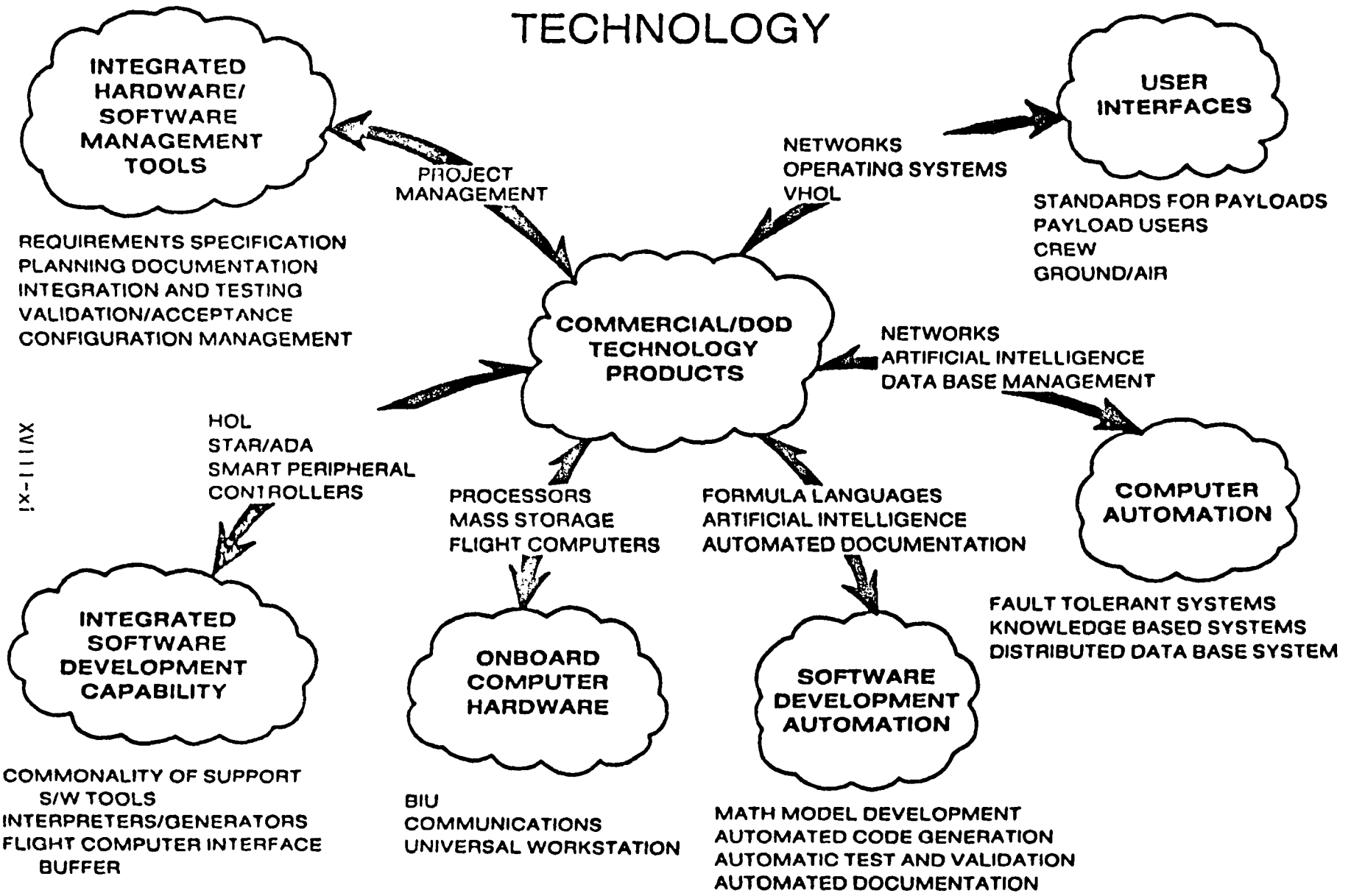


ILLUSTRATION 3

NASA-S-84-01364



XVIII-xi

ILLUSTRATION 4

5.0 INTRODUCTION

5.1 Purpose Statement

The SPACE STATION's design technology (deployment) is the next giant step in the coming decade of the United States' space program. Within the design and technology of the SPACE STATION will be computer capabilities which will operate with AI (artificially intelligent) expert systems which could utilize robotics, visual perception, voice synthesis; such as, speech synthesis and speech recognition.

These AI technologies have been predicted by industry analysts to account for 50 percent of all EDP (Electronic Data Processing) by the end of the 1990s--a forward direction for the computer hardware/software industry as well.

5.2 Natural Language Justification

The power of the hardware and computation facilities provided on board the SPACE STATION cannot be realized if there is not a user language available which gives the crew members free and flexible access to those resources.

The language must make full use of the on board facilities (which include the high technology development for on board software capabilities over the Space Station's life cycle) for the anticipated applications, while giving consideration to the possibility of commonality of the user interface with other user languages.

The crew members (previous programming experience not needed) must be able to interrogate the system to obtain answers, to monitor the system to determine performance, to create visuals to display screen dialogue and to trial check simulations, to obtain readouts to indicate status and diagnose failures, to examine databases for update, retrieval, and management, to exert manual control over applications for emergency management or for overriding systems software rationale, etc.

All of the above-mentioned crew-member tasks must be performed with ease without prior preparation, not to mention the other prepared-for and rehearsed crew member tasks.

What is needed is a natural; that is, English, French or Hebrew as opposed to FORTRAN, COBOL, Assembler, or BASIC, language that permits the untrained crew members to interact (access, query, update, etc.) with the underlying system in an extremely productive user-friendly manner. By user friendly is meant conversational, everyday English, no described syntax, etc.

The idea behind AI or logical processing is to make people more productive, and a natural language is perhaps the strictest and most difficult area of AI development.

5.3 Objectives

The SPACE STATION, on ground and in orbit, will require a Very High Order Language (VHOL) to query, produce and monitor performances of ground and on orbit tests.

The VHOL shall be structured with English type requests. The objectives of this research task are

1. Establish MSFC Mission Requirements to determine user language needs.

2. Review with other centers (KSC, JSC, GSFC) their activities in these areas.

3. Determine and design an analysis plan to develop MSFC requirements for VHOL.

4. Survey present VHOL commercial language.

5. Identify in-house capabilities presently available and determine future needs for on-going evaluations.

This research is related to the following MSFC Mission Assignments: SPACE STATION user language for use in test beds and inclusion in module development.

6.0 IDENTIFIED COMMON MODULE COMPUTER TASKS

The very high order natural language to be used by Space Station crew members must be able to interface with prewritten ADA, C, FORTH, FORTRAN, HAL/S, LISP, PL/1, Pascal, PROLOG, ASSEMBLER, etc. program applications.

This user-friendly, interface language will have built in AI applications.

These programs will be required to support such user operations as:

-Monitoring and Reporting Observations
-Examining and Updating System Databases
-Creating and Describing Graphic Displays of Picture Images' Innerconnection
-Monitoring On Going Systems and Making Adjustments Where Needed
-Troubleshooting On Going Processes
-Repairing and Recovering Satellites
-Identifying Extremely Dangerous Power Failures
-Detecting and Diagnosing Equipment Malfunctions
-Examining Realistic and Complex Situations and Making Judgments Based on these Situations
-Interpreting Data
-Testing Hypotheses
-Validating System's Performance
-Examining Programs' Performance
-Analyzing Stress on Internal and External Structures
-Avoiding Space Accidents and Correcting Problems in the Process
-Assessing Problems to Demonstrate Prototypes (Simulations) and Feasibility
-Controlling and Monitoring SPACE STATION's Atmosphere and Life-Support Systems
-Automatically Correcting Present and/or Predictable Failures
-Intervening or Concurring with Automated Machine Actions
-Interacting with Real Time Processes
-Forecasting Potential Conditions and Making Contingency Analyses

7.0 RECOMMENDED USER LANGUAGE STRUCTURES AND SYNTAX

There is justification and needful purposes for the development and/or incorporation of a Natural User Language (NUL) to be utilized by SPACE STATION crew members in accessing, querying, and retrieving information from each module's common computer databases, operating systems, expert systems or conventional applications. Illustration 1 (SPACE STATION Labs Initial Configuration) is Boeing's concept of the location of the computer in the common shell of each module. Note specifically the Workstation (in mockup) area of the illustration. Also of interest is Boeing's commitment to AI (Speech Recognition and Speech Synthesis) as well as Screen and Keyboard Applications.

7.1 Syntax

In order to accomplish the purpose of easy access and retrieval of information thru the system, syntax or order of arrangement (grammar) should be a secondary consideration.

7.2 Semantics

Semantics (meanings of the words identified for use in the user's natural language) would have primary emphasis. The natural user language would concern itself only in a transparent way with syntax by identifying either the spoken, written, or selected part or parts of speech utilized with the parser, interpreter, or translator; i.e., an interrogation and/or readout from one of the on board instruments is made. He/She might say, "I need to know what the attitude control requirements are for a given experimental configuration."

The parser would use its language interpreter to identify the nouns (attitude control), action verbs (need to know), etc. to interface (activate) a program and output the results.

7.3 Structure

SPACE STATION

Space Station Labs Initial Configuration

BOEING

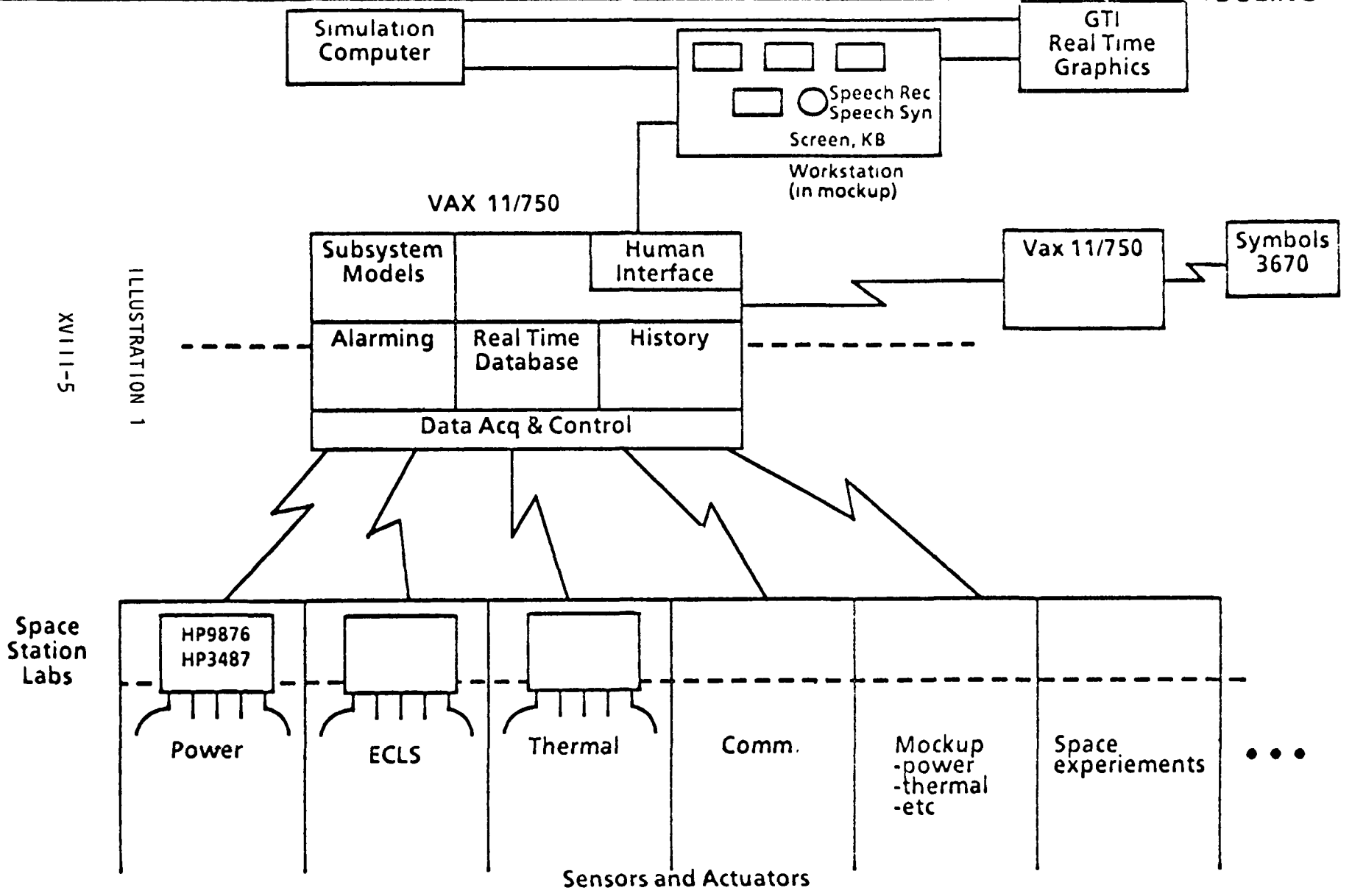


ILLUSTRATION 1

XVIII-5

The structure or form of this natural user language would be made up of prompts (statements from the system describing a specific action one needs to take in order for a task to continue).

7.3.1 Prompts

Prompts give the operator choices from which an appropriate response for the task one wishes to perform is selected.

Menus could be incorporated for use as displays of tasks to be performed or for initial user access to interface with ADA. Menus could display on the screen a list of tasks that the user can perform by choosing an ID letter and/or item description. Letters are used to identify the choices and descriptions or give details about each item.

By choosing from menus which help guide the user, the user tells the system what he/she wants to be done or indicates whether his/her speech synthesis and the computer's speech recognition are synchronized (which varifies and assures these AI processes).

7.3.2 Menus

Menus could be set apart as help menus, command menus, task menus, and procedure menus.

7.3.2.1 Help

Whatever user need arrises, there should be a help, command, or procedure menu to aid the accomplishment of the task or provide a message or statement which tells you about what has or is happening on the system and what response to make to get the appropriate action.

7.3.2.2 Tasks

The possible tasks that could be performed would be limited only by the firmware of the system and would be expandable depending on previous SPACE STATION experiences, payload applications, and scientific and technological advances.

Better still the very high order language (VHOL) Expert System could utilize real-time processing with menus and prompts to interact and execute interrogations, creations, indications, diagnoses, observations, adjustments, controls, description, etc.

7.3.2.3 Commands & Procedures

Task commands could also be placed on a menu with a letter or number code beside each listed task so that a selection could be made by the crew member which would in turn display a procedural menu with specific non-technical directions and/or prompted data entry requests responded to again by the crew member which would be interfaced with the task program for executing, reporting, updating, monitoring, adjusting, etc.

The task command menus would list all possible tasks common to all modules as well as tasks unique to the particular module's purpose. Programs which interface with these tasks could be written in any high order language acceptable or compatible to ADA or with whatever SPACE STATION language NASA selects.

7.3.3 Dialogue

The interaction between the crew members (astronauts) and the common module computer via the VHOL would result in a dialogue between human and computer understandable to both man and machine minus human knowledge base other than native language usage. In other words, it is anticipated that this natural user language be translated into the languages of the world in order that universal users of the SPACE STATION's technology and experimental capabilities might indeed be a cooperative (international) effort thereby advancing our goal of the commercialization of space.

7.4 ARTIFICIAL INTELLIGENCE

Artificial Intelligence (AI) has been defined in many ways. Notable among these definitions is the computer program that is knowledge based (logical) rather than digital in its handling of the data (numeric or text).

7.4.1 Expert Systems

Several programming tasks could be written in AI structure and format. These programs if they use knowledge and reasoning techniques to solve problems which normally require the abilities of human experts would be described as expert systems.

7.4.2 Robotics

Computers which have programmed applications which make manipulative (physical) motions (tasks) would fall into the AI Robotics category. Robotics could be invoked by crew members at night when they are sleeping instead of placing all their systems in the hands of ground control. If human intervention were required, it could alarm the crew members to take appropriate actions. Robotics could also be used on the arm which retrieves the satellites for repairs and maintenance as well as for the repairs made, thus relieving the crew members of this often frustrating, (crew members make human errors which consume time, energy, and fuel and make critical their second try), dangerous (crew members are assured of life for only eight hours when suited up in their 400-pound space suits), and human-expert (required) endeavor. With a working robotics expert system, crew members could successfully invoke the system with the VHOL.

Robotics would be used in the NUL's interface to make adjustments, to interact with real time processing, to intervene and to correct failures, etc.

7.4.3 Speech Recognition

Speech Recognition is described as the ability of the computer to understand the spoken word. Via the NUL (natural user language), the SPACE STATION crew members could give

verbal commands to cause the computer to translate the commands into program activation using speech recognition techniques. Before activating any programs, however, there would have to be varifying techniques built in so that the crew members could varify the computer's understanding of the voice commands.

7.4.4 Speech Synthesis

The computer's ability to develop spoken speech from text or other representations is an idea whose time to be tested has come.

If this AI approach were used for the NUL interface language, the computer's feedback would have to be monitored closely to insure reliability.

7.4.5 Computer Vision

Computers that reason, plan, and perceive based on visual sensory input, in which a symbolic description is developed of a scene depicted in an image, describes computer vision.

On the common module's computer, machine vision, or image understanding might be used in forecasting, tracking, creating and describing various SPACE STATION phenomena with the NUL being the interface to these knowledge-based, expectation-guided processes.

7.5 Functional Requirements

Depending upon the amount of AI utilization by the programmers who've written programs needing solutions as well as the yet-to-be-determined (YTBD) direction (AI or conventional) of the standardization and selection of one user language or the acceptance of a few user languages, specific functional requirements for the NUL will be defined.

If the direction is going toward AI, then certainly the functional requirements of the hardware would change from conventional hardware to hardware with AI capabilities, and the software interface NUL would undoubtedly be upgraded from CRT and keyboard input (menus, commands, prompts and feedback) to AI languages; such as, PROLOG and LISP and AI

programming tools; such as, ART, KEE and/or Language Craft which permit expert system programming to make use of speech recognition, speech synthesis, robotics, and/or visual perception for inputs/outputs.

7.6 Vocabulary (Lexicon) Dictionary for AI Applications

Here again AI input systems capabilities (parsers, interpreters, and language translators) would have to be incorporated as a required part of speech recognition expert systems which would handle the following:

7.6.1 Paraphrases--to give the meaning of a passage in other form or language.

7.6.2 Relative Clauses--a clause adjunct introduced by a relative pronoun referring back to a noun also linking a subordinate clause to a main clause.

7.6.3 Synonyms--one of two or more words of the language having the same or nearly the same meaning.

7.6.4 Pronouns--a word used instead of a noun; one of a small group of words referring to persons or things either named, asked for, or understood in the context.

7.6.5 Abbreviations--a shortened or contracted form of a word.

7.6.6 Antecedents--nouns or noun equivalents (whether word, phrase, or clause) referred to by a relative personal pronoun.

7.6.7 Acronyms--words formed from the initial letters or syllables of the successive parts of a compound term.

7.6.8 Idioms--expressions in the usage of a language that are peculiar to themselves either in grammatical construction or in having a meaning which cannot be derived as a whole from the conjoined meanings of their elements.

The lexicon dictionary would be limited to the paraphrases, synonyms, pronouns, abbreviations, acronyms, and idioms placed in the NUL dictionary and would be geared to prompt for another (other) word(s) by displaying a message indicating its level of understanding or would automatically display a tasks' menu it could perform from its understanding of the command given, so that the crewman could verify the computer's understanding with the correct response.

8.0 REVIEW OF POSSIBLE USER LANGUAGE INTERFACES WITH ADA

8.1 GOAL (Ground Operations Aerospace Language)

GOAL is a KSC (Kennedy Space Center) language design for shuttle integrated test checkout and launch operations, for payload integration with shuttle, and ESA (European Space Administration) Spacelab operations.

--Host/Target Machines: Honeywell 6680/Modified Modcomp IV and special purpose control console, keyboard, and graphics display

--Potential Applicability to Space Station NUL include: Language syntax and functional design meets requirements for English-like readable language for text and routine operational procedures; additional design work needed to develop short form of language for real-time, interactive operations; and re-implementation from language specificiation would be needed for new host hardware.

--Strengths: Language well matched to the process it is designed to control. Easy to learn English-like syntax provides readability and self-documenting procedures which aid in verification, validation, and maintenance.

--Weaknesses: High level of dependence on Shuttle ground computer hardware, not implemented with an interpreted/on-line interactive mode (compiled), limited application program interface and display hardware dependent.

8.2 JSC Prototype Crew Workstation for Space Station Environment:

Serves three classes of users: command and control (station maintenance), proximity operations, and potentially payload operations:

Primary Graphic Interface: Menu selection of predefined procedures and subsystem parameter/status displays selected by cursor position on system schematic,

includes concept that user must provide graphic display sequences and control software for all payload sequences in conventional programming langauges.

Limited prototype coded in assembler on 16-bit minicomputer.

Potential Applicability to Space Station NUL:

1. Mock-up is a potential user demonstration and evaluation facility for both graphical and command interfaces.

2. Mock-up can be used to study transition between and combined use of displays and command language.

3. NUL could be used to provide command interface mode and develop additional pre-defined procedures for mock-up.

4. Would require:

   a. Redevelopment on larger computer in high level language.
   b. Design of interface between NUL and display control software.
   c. Development of utility software for user display development.

8.3 Language Craft

Language Craft is the first and so far only commercial environment for constructing NUL interfaces that exploits the principle of caseframe instantiation and its methods of integrating semantics and symbol knowledge.

A natural language interface frees users from the cognitive burden of determining how to formulate commands or desires in artificial, unfamiliar, command or query languages. Thus, users can focus their attention on the underlying task, rather than on the medium of communication. Natural language interfaces enables users to make productive use of a system from the outset, rather than going through lengthy and frustrating familiarization rituals with each new system encountered.

From the end user's point of view, Language Craft prompts for natural language input, the application replies, Language Craft issues another prompt, and so on. Sometimes

Language Craft may find a user's input ambiguous. In this case, it paraphrases its alternative understandings back to the user and asks for a choice between them. Other characteristics of Language Craft the user may notice include spelling correction and ellipsis (filling in information missing from the user's current input from the context provided by previous input (relative clauses)). In addition, Language Craft handles yes/no and WH (who, what, where, when, and which) questions.

Language Craft supports two quite differnt types of user:

--The end-user of a Language Craft interface to an application.
--The developer of a Language Craft interface to an application system; i.e., the person who specifies the information that Language Craft needs to provide an interface to a specific application system. This person is called a *grammar writer*.

Natural language interfaces provide friendly and effective communication to a wide variety of software systems and utilities including databases, expert systems, electronic mail, operating systems, computer-aided design, and office automation software.

Given the desirability and widespread utility of Natural Language Interfaces (NLI), the question arises of why all software systems do not already incorporate NL front ends. The language comprehension task is far more complex than originally envisioned. Only in the past decade have NL researchers been able to build automated language analyzers. And very few of them are sufficiently robust and efficient to form the basis of a commercial quality product. To date, there exists no universal general-purpose language comprehension system capable of reading a textbook and conducting unconstrained conversation with humans. Nevertheless, language interfaces for well-defined tasks have been built and have proven quite successful.

Constructing a NLI has especially required several person-years of effort by highly-skilled computational linguists and AI researchers. In order to reduce the human resource requirements, the idea emerged of creating an environment to support the development of NLIs--an environment that would minimize the need for specialized AI skills. Perhaps more importantly, such an environment would cut down the development time required to produce a functional

natural language interface from several person-years to a few person-months.

Language Craft (LC) is implemented in Common LISP and currently runs on Symbolics 3600 LISP machines and DEC VAX operating systems. The LC environment is portable to several standard AI workstations and a variety of LISP machines.

8.4 LISP (List Processing Language)

LISP (List Processing) (1957) is the most widely used AI language in the United States. AI languages differ from conventional languages; such as FORTRAN or C, in that they concentrate on manipulating symbols and defining relationships. Moreover these tasks can be accomplished easily without concern for data handling and memory allocation. Because of this ease and flexibility in symbol handling, AI languages are much better at tackling unpredictable situations. And AI languages can be easily altered or expanded.

LISP is made up of thousands of functions and users can add their own. These new functions are then treated as those already defined.

AI languages such as LISP also come with many functions for writing, editing, and debugging programs. Windows divide the screen into several areas to show the program, results, and data simultaneously. As a result, LISP programs can be easily altered. Moreover, LISP can be interpreted to speed programming or compiled to increase performance.

Many versions of LISP (called DIALECTS) have been created. Some DIALECTS are written on top of standard DIALECTS, optimizing the use of the language for specific applications. But the disadvantage is that a program written in one DIALECT may not run on a machine supporting another DIALECT. Standards have been created in an effort to overcome the drawbacks associated with many DIALECTS, with COMMON LISP receiving much attention due to the backing by the Federal Government. There is word currently that Common LISP will try to set a more uniform standard for LISP programs.

Recent improvements in computer technology, together with the availability of some of the finest software development environments in existence have made LISP a

viable language. The protocol support Symbolics, IBM PC XT, and DEC VAX link LISP to their machines. Language Craft, ART, PROLOG, MACSYMA, ESDT, etc. are implemented in LISP.

LISP offers a complete set of tools for productive and creative software development. This environment makes exploratory programming and rapid prototyping practical and allows for the easy creation and maintenance of large software systems.

The LISP (or LISP Subsets) language offers extensible data structure for maximum flexibility in data representation and manipulation. Text and data structures are represented in identical manner so programs that generate other programs can be easily written. The programmer can write new constructs or even completely new languages and easily build them on top of LISP to handle a particular problem domain.

LISP has capabilities lacking in other computer languages in that it permits manipulation of complex structures and symbolic information. Because of its ability to manipulate symbols, it is used for expert systems, visual recognition, and other subcategories of AI. Its major drawback, however, is that its large size and flexibility can easily swamp a conventional mainframe.

New computers have been designed specifically to provide for an interactive environment in which both data and functions coexist and can be inspected or modified easily. Functions can be tested as they are written, and problems found quickly. These features have made LISP the preferred tool of AI researchers.

Symbolic processing allows computers to deal with complex knowledge and data in such a way that it appears to mimic human intelligence. The LISP computer language evolved to handle the constructs of symbolic processing. It now enables computers to be easily programmed to represent objects and the realationship among them.

8.5 PROLOG (Programming in Logic)

The Japanese announced that they had chosen logic programming and PROLOG for their ambitious long-term nationwide effort known as the Fifth Generation Computer System (FGCS) Project thus increasing the interest in the language.

Among the expressed objectives of the FGCS project is the development of fast, intelligent computer systems with the following capabilities: human-like decision making and learning, natural language and voice I/O (input/output), automatic program generation, and distributed processing.

In the FGCS project, logic programming is envisioned as the link between the fields of software engineering, database systems, computer architecture & knowledge engineering. It is to be used for problem specification and transformation, unifying functional programming and relational databases, developing single-assignment languages, and constructing rule-based expert systems and natural-language processors.

8.6 SCOL (System Control and Operations Language)

A need existed for a standard high-level interactive, user-oriented language to control system level testing and operation. The system test and operation tasks, written in a language other than SCOL (e.g., ADA, ATLAS, FORTRAN), would be responsible for performing complex and time-critical actions and analysis, system monitoring and test instrument interface.

SCOL shall be used in a spacecraft system test environment. Precompiled software may be in Assembly, FORTRAN, ATLAS, PASCAL or any language a user selects. SCOL could reside in a real-time computer system used for control and monitoring of spacecraft system operations. Precompiled software performs telemetry processing, command processing, and display processing.

The language is to be defined using a formal syntax notation and syntax diagrams. The syntax definition shall be supplemented with semantic rules and constraints. All definitions shall be consistent with internationally recognized standards. The language and its implementation could provide generality only to the extent necessary to satisfy the needs of system level testing and operation.

The language should be designed to avoid error prone features and to maximize automatic detection of programming errors. The language should be easy to learn. Implementations should be easy for experienced and inexperienced personnel to use. The language and its implementations should be designed to facilitate rapid interactive use of terminology familiar to system control

and operation personnel. The language will provide a user-friendly tool for complete control and monitoring of the system.

8.7 SMALLTALK

SMALLTALK created by XEROX's Palo 'Alto Research Center (PARC) has gained prominence recently because of its ability to handle objects rather than functions. In this approach, a string not only has a value but a range of characteristics as well. These characteristics are "inherited" from the class that an object belongs to unless an exception is defined. Messages are used to perform operations on objects. Object-oriented programming can also be implemented in LISP by a set of functions; known as FLAVORS.

8.8 SSOL (Space Station Operating Language)

SSOL is an automated environment friendly to users, in which SPACE STATION Integration and Tests (I&T) Activities can be designed, developed, tested and performed. The SSOL System is being developed by KSC (Kennedy Space Center) and will be responsive to users independent of their location. Also, it will promote standardization and transportability of user-developed procedures from site to site, capitalizing on the functional commonality of the I&T activities at these sites. Advances in the user-friendliness of commercial real-time operating systems, data base management systems, supporting software development tools, processing speed, and memory capacity are only a few of the anticipated technological improvements which may enhance the SSOL System.

An analysis has been performed to determine the validity, integrity, and completeness of the initial SSOL System concept. The results of this analysis, and the emphasis placed on the need for an I&T concept of automation, were used to derive and refine SSOL System requirements and concepts.

The SSOL System is both a ground and onboard SPACE STATION Data System (SSDS) service.

8.9 STOL (Standard Test & Operations Languages)

This language is used extensively at GSFC (Goddard Space Flight Center) and Laboratory for atmospheric and space

physics (LASP) and implemented on DEC Systems PDP 11/70, PDP 11/34 and VAX machines.

Strengths:

1. Transportable--90% of code is FORTRAN IV Plus
2. Designed for interactive use in real-time operations
   a. Processed via language interpreter
   b. Responsive to user needs in real-time environment
   c. Easy to learn
   d. Can generate command procedure lists callable by name
3. STOL was designed to be and has been successfully extended
4. Data query commands for selected parameters, limits, state variables, and conversions allow little user modification

Weaknesses:

1. Slower than compiler languages
2. Readability compromised somewhat by concise command words
3. STOL is not under strict configuration control
4. "Single User" mode disallows multiple command stream generation required by HSC OPS (except for LASP SME-VAX version)
5. Terminal device I/O is operating system dependent
6. No graphics capability

Potential Applicability to SPACE STATION VHOL:

Storing "pre-interpreted" procedure lists appears feasible and would provide speed advantage. Readability can be improved by incorporating front-end translators. Extensions required to provide additional data query functions. Performance good for real-time operations, not necessarily for the critical or concurrent process control functions.

8.10 TAE (Transportable Applications Executive)

TAE was developed at Goddard Space Flight Center (GSFC).

Environment:

--Payload/instrument data analysis

--Provides standard interface:
  -to user for application program control and parameter definition
  -to application program for user interaction and operating system services
--In wide use in NASA and universities
--Originally developed for VAX under VMS. Posted to: VAX under UNIX, PDP 11 under RSX-11M and Data General under RDOS.

TAE - Interface Modes:

--Menus - Tutored Input Mode
--Command Mode
--User Created Procedure with
  -Global and local variables
  -Conditional statements
  -Looping

Strengths:

--Effective user interface for infrequent users (menu mode) as well as expert users (commands and procedures).
--Provides common interactive user interface to applications programs. Support System extendability.
--Highly protable - 87% of code.

Weaknesses:

--Interactive but not designed for real-time - some speed penalty.
--No graphics interface.
--Not in use for either of primary UIF (User Interface) functions - I&T or real-time operations.

Potential Applicability to SPACE STATION UIL:

--Use as prototype for user feedback on limited subset of language features (e.g., TAE tutored Input and Command Modes, approach to procedure capability).
--Design concept of common user interface software with application program interface routines for parameter acquisition.

8.11 Third-Party Software: such as, ART, KEE, & MACSYMA

ESDT (Expert System Development Tools) are programs that aid the user or knowledge engineer extract and code information required to build and test and expert system.

ART (Automated Reasoning Tool) is a comprehensive software tool that basically turns engineers into builders of full-scale expert systems. ART was designed as a tool for developing expert systems, programs that combine human expertise with artificial intelligence (AI) to produce results that have been beyond either computers or humans in the past. It contains an inteference engine, which resembles an operating system of conventional systems; a knowledge base, which replaces a data base; an editor, which helps create the knowledge base, and a monitor, which lets users visually follow the tool's reasoning steps.

Users can interact with the ART system quickly and easily--the novice, through hierarchial menus, and the experienced engineer, directly through the keyboard. Software mechanisms automatically correct errors, and graphic representations of the run-time knowledge base let the user visualize ART's approach to a problem immediately.

The world of AI deals with reasoning through forward and backward chaining. The first uses goal patterns to cull new facts from existing declarative knowledge. The latter depends on strategy patterns to draw simpler goals from more complex goals. Forward chaining moves from the data toward the overall goal; backward chaining attempts to successively reduce the overall goal until it finds matches in the known data. Both methods in chaining are valuable procedural organizations for expert systems.

ART reasons through both types of chaining; rules may involve both goal and strategy patterns. ART automatically constructs and maintains a viewpoint structure that represents all hypothetical possibilities an expert system is considering. Rules can explicitly compare and contrast the possibilities to decide on the best possible solution.

The expert systems that ART produces are written in LISP.

KEE (Knowledge Engineering Environment) (IntelliCorp - 1983) was the first ESDT commercially available "Tell and Ask" language which helps define and retrieve facts from the knowledge base in sentences. Active images permit users to create graphical displays for viewing and controlling the system. KEE is run on all AI machines and is being applied to a number of engineering and manufacturing situations, including simulation of production facilities.

MACSYMA, developed at the MIT Artificial Intelligence Laboratory and marketed by Symbolics, Inc. of Cambridge, MA is one of the earliest and still most impressive AI based software packages. It is an intelligent system that uses the flexibility and power of symbolic processing to solve equations of algebra and calculus so complex that they are generally beyond the skill and patience of human mathematicians. MACSYMA produces important analytical solutions that could otherwise only be approximated by numerical methods. Moreover, computation results can be printed out as more easily comprehended, and often as beautiful graphic presentations. Indicative of the system's power is the fact that a single line of MACSYMA code is typically equivalent to eleven lines of conventional FORTRAN.

MACSYMA's ability to compress entire mathematical entities like variables, symbols and operations into symbolic objects points out one of the major commercial advantages symbolic processing has over numeric processing. Applications software can be created faster and less expensively using symbolic programming methods. Because data and commands are separate, programs can be developed, prototyped, tested, and edited in more efficient increments.

MACSYMA is an "expert system." It provides logical rules which use facts in the data base to infer other facts about how to solve an equation. As in all expert systems, the secret of MACSYMA's power is in its data base.

8.12 UIL (User Interface Language)

UIL is a set of software tools for a flexible but standard user interface to the SPACE STATION System, SPACE STATION Payloads, and Platforms which supports on-orbit plus ground integration, tests, and operations, for use by engineers, crew and scientists. It provides English-like language for familiarity and readability; short form for real-time interactive control procedure capability; includes software utility support for graphics, display, data, and dialogue interface and control. Users would include: Development, test and checkout engineers, launch operations engineers, ground mission operators, flight-crew, flight and ground payload/instrument scientists and engineers.

As a result of the UIL study, it was recommended that a standard, high-level language be developed to make phase and location differences transparent to users & user procedures.

9.0 VERY HIGH ORDER SPACE STATION NUIL SUMMARY

After having reviewed the President Ronald Reagan State of the Union Message of January 25, 1985 which directed NASA to develop a permanently manned SPACE STATION within the next decade, NASA began to divide the task into digestible parts. The NASA/MSFC EB Lab's part evolves around the software evaluation of the common module computer cores.

Because of the various trained- and practiced-for tasks of the crew members, and because of the different levels of expertise of the crew, it was felt that the development of a user-friendly interface language would increase productivity of the crew members as well as encourage and increase commercial ground users. This NUL would be able to either prompt the users for responses or would accept their natural language for commands. Crew members or ground users could utilize the computer with or without having prior programming knowledge; however, users with programming knowledge would not be limited to the applications (expert systems) firmware either.

The system should be expandable over the 30-year life of the SPACE STATION. To envision what will be 30 years in the future is exhaustive, but by "sky is the limit" brainstorming, I can foresee expert systems that have language translators in the popular languages of the world (i.e., English, French, German, Spanish, etc.) which would accommodate Americans and our friends around the world in free enterprise, accept each users natural language input, and return each users natural language output by using speech recognition, speech synthesis with audible, display and/or hardcopy outputs.

This expertise would be evolutionary and would initially require tutored display, keyboard responses, or touch screen inputs for testing, monitoring, controlling, and maintaining the system and who knows what else this system will be expected to do in the future.

The next problem involved a review (evaluation) of what has been done in this area to see whether the expert-system program had to be started from scratch using a newly created language or whether an already existing interface language could be used or modified for use in this situation.

GSFC's, KSC's, JSC's, LRC's, and MSFC's current reports and expertise on the subject were reviewed, demonstrations and three workshops were attended, two long-distance calls were made to JSC (Sandy Richardson), and Prime Contractors, Clive A. Arlington - Martin Marietta Denver Aerospace and Dr. Byron Purvis - BOEING, Corp. were consulted, a class in LISP was taken, and John Wolfsberger, my boss, added to my knowledge base and contributed greatly to this study.

Some existing interface languages and AI programming tools were reviewed: GOAL, JSC, Language Craft, LISP, PROLOG, SCOL, SMALLTALK, SSOL, STOL, TAE, ART, KEE, and MACSYMA. From my review, I found that there were two, one already developed language and another AI programming tool which I felt needed further study for possible adoption for use in the SPACE STATION NUIL development.

Recommendation 1--TAE originally developed for DEC VAX under VMS by GSFC for the initial SPACE STATION Common Module UIL which utilizes all the user-friendly requirements, I envisioned as being needed initially (tutored inputs with prompted responses, menus, helps, and commands) for user program development, and program application execution.

Modifications and expansions would, however, be needed to increase its real-time speed and interactive abilities, to add graphic interface, and to include I&T and real-time operations which are primary interface functions.

Recommendation 2--Language Craft, an integrated natural language processing environment for developing NLI for operating systems, databases, and knowledge-based systems (which has a language parser which handles synonyms, paraphrases, pronouns, verbs, etc. needed for speech recognition and synthesis) be used to eventually convert a TAE Subset interface to a speech synthesis, speech recognition expert interface system and retain parts of TAE Subset for verification of spoken commands via prompted responses.

In the researcher's view these were the only two languages reviewed which would not require prior user programming knowledge. Also, previously written conventional language programs could use this NUL as transparent front-end software for easy user access.

10.0 REFERENCES

1. NASA Advanced Technology Committee (ATAC), NASA Executive Overview, NASA Technical Memorandum 87566, Advanced Automation & Robotics Technology for the SPACE STATION and for the U.S. Economy, Volume 1, March, 1985.

2. Mike Argabright, Chairman, SCOL Subcommittee of IEEE Atlas Committee, "SCOL (System Control & Operation Language) Requirements Specification," 1 October 1983.

3. Jaime Carbonell, "Language Craft Reference Manual (Limited Edition Release)," Version 2.0, Carnegie Group, Inc., 1985.

4. Clara Y. and John L. Cuadrado, "Prolog Goes to Work - What Prolog Is, Who's Using it, and Why?" BYTE, August, 1985, Volume 10, No. 8, pp. 151-158.

5. John Dalton (GSFC), Chairperson, "Report of the Ad Hoc Committee for SPACE STATION User Interface Language," July 9, 1984.

6. A. Dorofee and L. Dickison, "Space Station Operations Language (SSOL) System Requirements and Concept Definitions," August 1, 1985.

7. E. A. Feigenbraum, A. Barr, and P. R. Cohen (Eds.), The Handbook of Artificial Intelligence, W. Kaufmann, Publisher, Los Altos, CA, 1982.

8. William K. Hartmann, Ron Miller, and Pamela Lee, Out of the Cradle--Exploring the Frontiers Beyond Earth, Workman Publishing Co., NYC, 1984.

9. Abraham Hirsch, Program Marketing Manager, Symbolics, Inc., "Artificial Intelligence Comes of Age," COMPUTERS AND ELECTRONICS, March, 1984.

10. Abraham Hirsch, "Tagged Architecture Supports Symbolic Processing," COMPUTER DESIGN, June, 1984.

11. Robert B. Mills, "Building Your Own Expert System," COMPUTER AIDED ENGINEERING, June, 1985, pp. 76-78, 80, 84, 86, and 90.

12. Susan Voigt, Editor, NASA Language Research Center, "Report on an Open Forum with Industy and Academia," Held at NASA/MSFC, Huntsville, AL, April 24 and 25, 1985, NASA Scientific and Technical Information Branch.

13. Robert Wilensky, LispCraft, W. W. Norton and Co., New York, 1985.

14. Chuck Williams, "Software Tool Packages the Expertise Needed to Build Expert Systems," ELECTRONIC DESIGN, August 9, 1984.