N90-27300

# Space Station Advanced Automation*

Donald Woods
MDSSC-SSD
16055 Space Center
Houston, TX 77062
(713)280-1500
DWoods@NASAMAIL.NASA.GOV

In the development of a safe, productive and maintainable space station, Automation and Robotics (A&R) has been identified as an enabling technology which will allow efficient operation at a reasonable cost. The Space Station Freedom's (SSF) systems are very complex, and interdependent. The usage of Advanced Automation (AA) will help restructure, and integrate system status so that station and ground personnel can operate more efficiently. To use AA technology for the augmentation of system management functions requires a development model which consists of well defined phases of: evaluation, development, integration, and maintenance. The evaluation phase will consider system management functions against traditional solutions, implementation techniques and requirements; the end result of this phase should be a well developed concept along with a feasibility analysis. In the development phase the AA system will be developed in accordance with a traditional Life Cycle Model (LCM) modified for Knowledge Based Systems (KBS) applications. A way by which both knowledge bases and reasoning techniques can be reused to control costs is explained. During the integration phase the KBS software must be integrated with conventional software, and verified an validated. The Verification and Validation (V&V) techniques applicable to these KBS are based on the ideas of consistency, minimal competency, and graph theory. The maintenance phase will be aided by having well designed and documented KBS software.

## Introduction

The development of complex space systems is a costly endeavor; however, the operation of these systems is where the majority of the cost will occur. For example in the National Space Transportation System (NSTS), approximately 50% of the current program cost went into the design; however, by the end of the program lifecyle operations will have accounted for 95% of the total program cost. The important thing to remember while designing the system is that the ease of operations and maintenance will be the long term life cycle cost drivers.

Building the space station will be more difficult than the shuttle. For example, it will exist in at least 20 different configurations during the assembly phase, and at each of these minor milestones, it must meet different requirements with different resources while still maintaining safety. The shuttle was built on the ground by one major contractor while the station will be built in space by 4 NASA work packages, and 3 international partners.

All of these factors taken together point out that the space station will be one of the most complex engineered systems ever taken into space. The Systems Engineering and Integration (SE&I) in the program are of utmost importance. A&R has been recognized by congress, NASA, and the contractors as being an enabling technology for designing a safe, efficient, reliable, and maintainable station. However, KBS have no well defined standards for evaluation, development, and integration. Possible technical approaches to be employed in each of these tasks will be explained. Without well defined engineering approaches few engineers will be willing to give this technology a chance in a budget, and safety conscious environment.

* Work funded by NASA contract NAS 9-18200, the technical monitor was Jon D. Erickson.

## Motivation

The operation of the shuttle requires approximately 5,000 people to support a 10 day mission every 90 days; the space station will be onorbit for 30 years, hopefully this won't require 45,000 people to support. The operation of complex systems requires a lot of manpower. The traditional approach to dealing with faults has been to develops malfunction procedures based on Failure Modes and Effects Analysis Critical Item List (FMEA/CIL). This approach has been fairly successful in dealing with common failures, with a few exceptions. For example out of 9 inflight Shuttle Remote Manipulator System (SRMS) failures none were covered in approximately 200 FMEA/CIL sheets. We are able to deal with the failures we anticipate; however, the unexpected failures resulting from faulty instruments, and unexpected causality are very hard to isolate. Currently in the NSTS program these faults are isolated by ground controllers who examine telemetry data, check system schematics, execute simulations in the mission evaluation room, and basically do very difficult and thorough analysis in real time.

These demands upon the controllers have caused them to investigate, develop and use a KBS. The first KBS in the Mission Control Center (MCC) is the Integrated Communications Officer (INCO). It has been so well accepted by the controllers that at least four more KBS consoles are being planned: Guidance, Navigation, and Control (GN&C), On-orbit propulsion, Electrical Power, and Life Support [1]. INCO has produced major cost savings ($480K/Year) [2]. The personnel at Kennedy Space Center (KSC) have also identified A&R as being one of the most important new technologies which should be developed for SSF [3]. It is clear that the personnel involved in the operation of space systems consider automation very important, and even necessary.

## Development Model

In the automation development model (Figure 1) several distinct phases are identified which will insure that safe, productive, and reliable automation applications are put into the SSFP [4]. Steps 1 to 3 help define what is clearly needed in terms of requirements, and the various changes that constantly occur in major programs. The goals and objectives (step 4) are used to develop criteria for candidate selection. The approaches of the subcontractors in meeting these requirements is feed into the A&R management function where a decision is made along with the customer on whether or not a function is worth evaluating for eventual inclusion into the baselined system. A detailed assessment of the support capabilities available in the program will consider the maturity and availability of KBS tools in the SSE, analysis of the Master Measurement and Command List (MMCL) for sufficient sensors and actuators to allow automated operation, inter- and intra system dependencies, and what KBS technology is applicable to the problem.

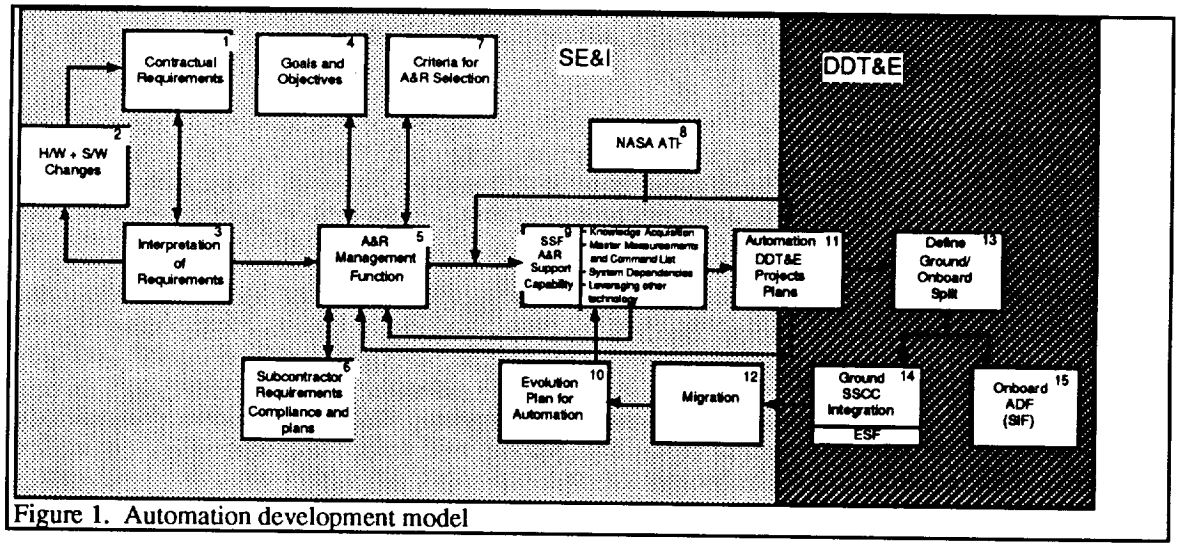

Figure 1. Automation development model

During Phase B a thorough evaluation of the Space Station's (SS) functions identified 22 candidates which could be automated. Four different approaches were considered for doing these candidate evaluations: linear weighting, fuzzy set method, Weight Outranking Method (WOM), and Multi-Attribute Utility Theory (MAUT) [5-6]. The last two were used. MAUT requires a greater degree of definition of the

system, and each candidate evaluation is independent of other candidates; however, it is generally more accurate if criteria and relative importance are well defined. WOM causes each candidate to be relative to each other candidate, and with a large number of candidates ranking can be difficult. If any further evaluations must be performed MAUT would probably be the more useful measure considering the level of design detail at present. This sort of analysis can also be used to determine the most applicable KBS or traditional technique for implementing a given function. In many cases it will probably be a combination of both. Based on this work there are two broad categories for which automation is being considered for performing SS functions: Planning and Operations. Planning entails resource, logistics, and maintenance management and scheduling. Operations involves: monitoring and analysis, Caution and Warning (C&W) filtering, Fault Detection Isolation and Recovery (FDIR), Testing, Fault Tolerance and Redundancy Management (FT/RM), and Command and Control (C&C). This analysis is also justified as part of the required logistics activities; for example in MIL-STD-1388-1A, the following tasks are identified: 301, 302, 303. Task 301 defines the requirements; task 302 identifies possible technologies for implementing these operational requirements; and task 303 performs the trade-study to identify the most promising implementation technique.

The next step is the development of a Design, Develop, Test and Evaluation (DDT&E) plan. This is where the focus of activities moves from SE&I to the responsible development organization (usually avionics/software). The main thing that most KBS should emphasize is that they should augment, not replace existing systems, and be initially used in an advisory mode. When used in an advisory mode the system can be evaluated with little risk.

Results of the DDT&E prototyping efforts will be used to assess the feasibility of onboard implementation. Recommendations relative to placement of functions on the ground and onboard will be documented. If the function is designed to go onboard then it will enter full scale development with the Avionics Development Facility (ADF) as its target. For that part of the application which is designated as ground software, an analysis will be conducted to determine hooks and scars, and a migration plan will be developed that is consistent with program guidelines.

### Life Cycle Model

The waterfall life cycle model is the most commonly used methodology for development of software (DOD-2167, and the Software Management and Assurance Program (SMAP) 3.0). In the past KBS have been developed using "rapid prototyping", "iterative" or the "add rules till it works" development model. This method does not lend itself to requirements traceability, verification, validation, maintainability or reliability. This "iterative model" should not be confused with the spiral approach which is driven by risk reduction, rather than being "code driven". In a situation such as the space station program where the hardware and software are being developed simultaneously it should be possible to develop KBS using the flexible life cycle model allowed by SMAP, version 4.3. The most important thing this model allows is a period of controlled prototyping to define requirements. This pre-requirements "prototyping phase" is controlled, meaning that goals are defined a priori, and the results are documented in the appropriate documents. Four sub-phases will be repeated until the Analysis sub-phase fails to identify new tasks or problems requiring further prototype development. The four subphases are: analysis, knowledge acquisition, design and implement, and test and evaluation (figure 2).

During the analysis phase Software developers will determine the scope of the problem and the form of the probable solution(s). In later iterations, this analysis will be partially provided by testing from the previous pass. The product of this sub-phase will be a set of tasks which the prototype is required to perform at the end of the iteration. The knowledge acquisition phase will collect the knowledge that is required to carry out the tasks. A description of this knowledge will be the product of this sub-phase. In the design and implement sub-phase software developers will select algorithms, search routines, paradigms, and knowledge representations that are necessary to produce solutions from the knowledge acquired in the previous step. The series of design decisions and the final design solution are the product of this sub-phase. During the test and evaluation sub-phase software developers will design tests to verify that the prototype system is behaving as desired. The product of this sub-phase will include descriptions of the tests, and interpretations of the test results. If the prototype system is performing suitably, then its current functionality will be transmitted to the appropriate phases of the SMAP development model.

All other SMAP mandates and guidelines will be observed. Specifically, Verification and Validation of expert system application software will be incorporated into all phases of software development. Also, expert system application software will conform to the SMAP standards and practices with respect to: application software integration, simulations, testing, delivery, interfaces, and acceptance.

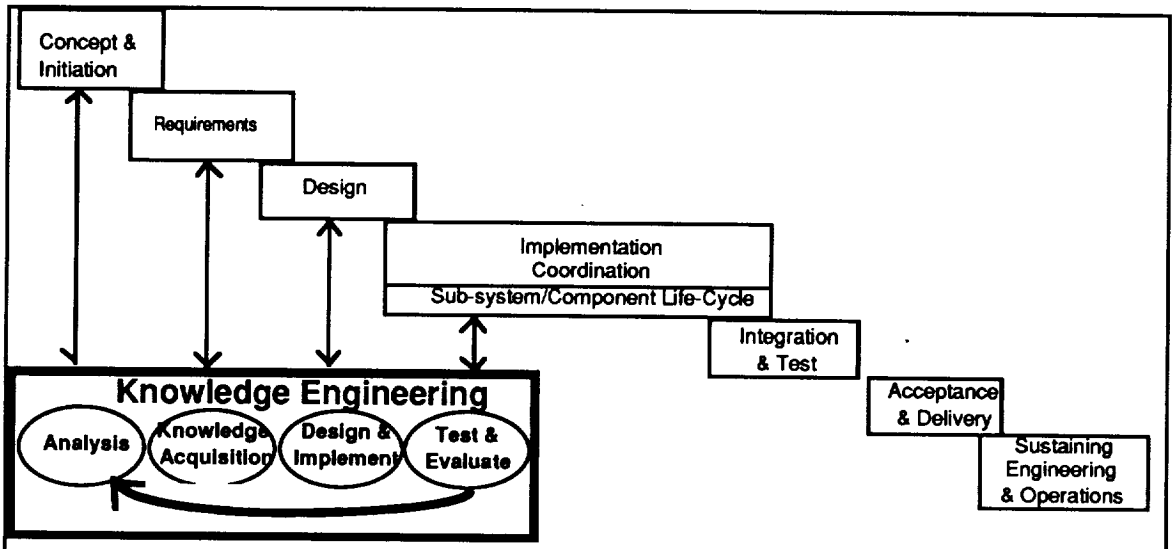

Figure 2. Modification of the SMAP LCM for development of KBS.

Given that SMAP 4.3 defines an acceptable life cycle model, the next major capability which needs to be in place is an acceptable KBS tool which will work within the capabilities and constraints in the SSFP: SSE, Ada, and SMAP. In the past most KBS products have been developed on symbolic processing computers using LISP. While at one time serious consideration was given to developing space qualified symbolic processors; this effort has been suspended. With the availability of KBS tools in ADA (CLIPS, ART and TIRS) this has become somewhat a moot issue; however, the integration of these tools into the SSF Data Management System (DMS) architecture is an open issue.

### Systems Engineering

The space station has highly coupled and interdependent systems. Realizing that no one system can or should have to maintain a world model, the concept of the Operations Management System (OMS) was developed. The OMS consists of an onboard portion, the Operations Management Application (OMA), and a ground based portion Operations Management Ground Application (OMGA). The basic function of the OMS is to carry out the Operational Short Term Plan (OSTP) taking into consideration the stations current state (resources, and constraints). The OMS is also responsible for station wide FDIR. While each system is required to do internal diagnosis, and may for time or safety critical functions reconfigure autonomously, the OMS has to be responsible for determination of across system impacts, and final determination of the best reconfiguration. In the operation of the shuttle once one failure occurs, the immediate question is what is going to fail next; the OMS should help with that analysis. Figure 3 shows the relationship between the OMA and the systems. The OMS may also have to deal with multiple automated systems using KBS techniques. If that ends up being the case then some type of fusion will be needed; one popular KBS technique is that of a blackboard [7]. Blackboards provide a global data base for multiple expert systems to access asynchronously, and allow cooperative problem solving. The theory of systems [8] should be used in the development of automated systems in a much more methodical fashion, for example decomposition, criticality analysis, information requirements analysis, and function and requirements allocation tools should be used to help define well engineered automated systems [9]. How systems theory can be applied to fault diagnosis is covered in [39].

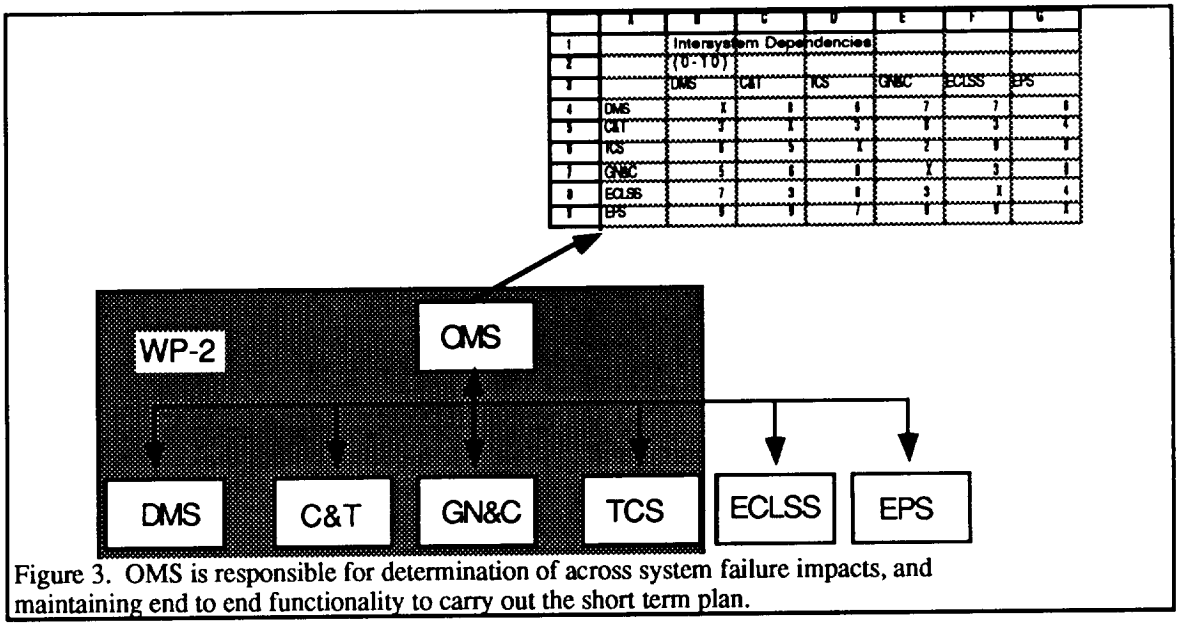

Figure 3. OMS is responsible for determination of across system failure impacts, and maintaining end to end functionality to carry out the short term plan.

**Software Engineering**

Just as object oriented programming principals (i.e. software engineering) have popularized the ideas of abstraction, objects, maintainability, and reusability, it appears that KBS S/W can also be engineered to have the same desirable characteristics. There are two sorts of KBS reuse: knowledge reuse, and reasoning (or inference) reuse (see Figure 4). Once the knowledge has been acquired in a useful form it can be used for a variety of applications such as Fault Detection Isolation and Recovery (FDIR), Intelligent Computer Aided Training (ICAT), Planning and Scheduling (P&S), and operations advisors. Once a way of doing any of these applications is determined, it is possible to use the same inference mechanisms in a variety of systems. Model Based Reasoning (MBR) has popularized this idea for FDIR, for example fluid, heat and electrical flows are all similar and obey the same basic physical laws: Kirchoff's current and voltage laws (i.e. conservation of energy). So once you know how to deal with a open or closed resistor by extension you know how to deal with a stuck open or closed valve in a fluid system. MBR also brings a host of other useful tools to the table, such as constraint propagation, and suspension, and qualitative modeling [10-19, 41-42]. The biggest open issue in MBR is what level of modeling is best suited for a particular problem. The simulations being developed fit into 4 categories: 1.) highly accurate mathematical model with real dynamics, 2.) very accurate model, but without all of the coupling between elements, 3.) table driven interface checking, 4.) very low fidelity. One of the big arguments with model based reasoning is that it is computationally excessive, for example the Thermal Expert System (TEXSYS), which implemented DeKleer's algorithm [18] for dealing with multiple faults, required 155,908 Source Lines Of Code (SLOC), and a variety of computers (symbolic and traditional) to perform the monitoring and control of the thermal testbed. While what TEXSYS accomplished was impressive (FDIR of 7 system level faults, and 10 component level faults), it should be considered as only having scratched the surface of what MBR can accomplish.

In almost all cases the most useful knowledge representation for physical systems is one based on structure and behavior. Such a model is much more flexible, and multipurpose than rules, decision trees, or a frame based system. Qualitative modeling can view components at a variety of resolutions, for example in electronics: a diode can be considered from the following viewpoints: as a binary switch, or as a linear approximation, a nonlinear approximation, in terms of electromagnetic fields (solving Maxwell's equations), going to the atomic level and considering quantum mechanics, or considering relativistic effects (quantum electrodynamics). In most cases the first two models mentioned are sufficient. Qualitative modeling also allows one to view systems (connected components) at different levels of detail, for example an operational amplifier can be considered in terms of inputs, and outputs voltages, and amperages, or as a differential amplifier. One is a result of the physical makeup of the device, the second interpretation results from the fact that it is an engineered device designed to perform a function.

**Automation Integration**

Besides architectural compatibility and interfaces the major integration efforts for KBS must be in the area of Verification and Validation (V&V). Traditional S/W V&V is largely a heuristic art that involves path checking and test generation. The two main methods are the black box and glass box methodologies. The black box is the interface checking functional requirements and limit checking verification, while the glass box is more concerned with how the functions are accomplished, exercising all branches, and code walk throughs. The importance of modularizing a program or rules can not be over emphasized, without it one is faced with a combinatorial explosion of possible paths.

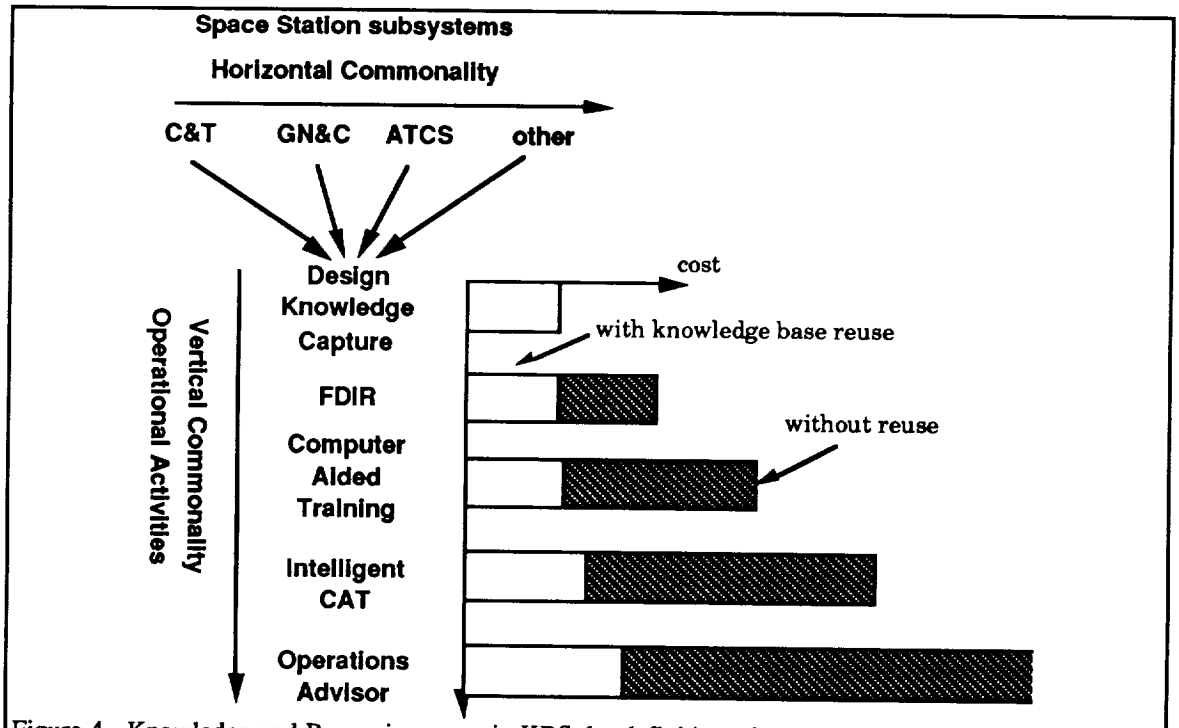


Figure 4. Knowledge and Reasoning reuse in KBS, by definition of vertical and horizontal commonality can control development costs.

Current research gives many clear directions on how to do V&V of rule based systems [20-26]. Rushby [25] develops the idea of "minimal competency" which states: while an optimum solution may be hard to define or verify, it is possible to define quite sharply what the minimum level of acceptable behavior is. Nguyen's CHECK program [22] checks the consistency of the knowledge base in the following fashions: redundancy, conflict, subsumption, unnecessary if rules, circularity, unreferenced and illegal attribute values, unreachable conclusions and dead end goals or if conditions. Stachowitz [21] has extended these conditions to deal with certainty factors [23]. The only difficulty is the integration of these techniques into a tool which is usable in the SSFP. One of the biggest stumbling blocks in the way of using rule based systems is that these V&V concerns are all trying to address the basic problem that rule based systems are not bounded in time and space computationally because of the way the underlying data structures are setup (Rete or Treat networks).

Since decision trees are derived from data, the data must be accurate; however, generating Ada code from a decision tree is straightforward, and therefore the V&V should be easily accomplished using Ada V&V methods. Model Based Reasoning can be done in procedural languages, so ADA should provide no problem however the lack of tools does. The issue of V&V for MBR systems is basically to make sure the simulation used is accurate. This simulation can be considered the knowledge base. The reasoning techniques which interface with these simulation knowledge bases should be verified and validated separately (just as rules and their interpreter in expert systems are validated separately).

The other issue involved in the integration of KBS is interfacing with Ada procedural code. Since there are a variety Ada KBS tools (CLIPS, ART, and TIRS) this should not be a major problem.

**Diagnostics**

Currently most of the space station's systems have baselined Built In Test/Built In Test Equipment (BIT/BITE) as being the solution to performing FDIR. The main advantage of this technique since it is performed in hardware, is its quickness. While this is a popular technique, the shortcomings of this approach are fairly well known: false alarms, can not duplicates, retest okay, and failure to diagnose. In fact the unreliability of BIT/BITE has prompted the generation of measures of how poorly it performs [27], as noted "the experience with automatic detection and isolation systems, in the form of BIT, has not lived up to expectations. False alarm, false isolation, and failure to diagnose errors are reported as a result of system diagnostic inadequacies. With the existence of such errors, the evaluation of the operational capability of BIT/BITE becomes a real challenge."

While BIT/BITE is not useless, it should definitely not be the only technique available for FDIR. A more intelligent approach which considers the overall systems state will help remove false alarms by verifying that each Orbitally Replaceable Unit (ORU) is receiving its necessary inputs with no noise or bad signals. It is extremely important that the inputs and outputs of each ORU are well instrumented to help with eventual automated diagnosis. Failure to diagnose errors in BIT/BITE can be solved by considering long term trends such as sensor failures and calibration errors (this is a 30 year not 10 day mission).

It would appear that using BIT/BITE by itself won't solve our problems. We could create a "fault dictionary" by using a simulation and a list of the kinds of faults anticipated. This can be considered an automated FMEA/CIL search table. This results in a list of fault/symptom pairs, we can then invert this list in a variety of ways: Bit string register level comparisons [28], machine learning [29], or even a rule based expert system [30]. The problem with this approach when used on any reasonably complex system (a space station for example) is that the creator of the diagnostics must settle on a small, fixed class of expected faults so that acceptable fault coverage (statistically likely), isolation, and computing efficiency can be realized. A fault is predefined, while it actually should be "anything other than the intended behavior" [12]. Again it should be emphasized here that this sort of diagnosis is good as a second layer on top of BIT/BITE to help with some of the ambiguities, and inconsistences that arise, but should not be the only answer.

Perhaps decision trees [31-33] could help with writing down an efficient sequence of tests and conclusions to guide a diagnosis. Decision trees are useful in other aspects as well; for example, they can help flag useless and redundant tests, and depending on the algorithm used to generate them usually give the most efficient sequencing of tests available based on information theory (entropy), but even this heuristic can be in error by as much as 30 percent from the optimal solution [32]. Again the point is that they are a way of writing down an answer which is already known.

Rule based expert system provide an efficient computational mechanism [34] for using the knowledge of expert troubleshooters. This approach is more intuitive, and perhaps more in tune with the actual operation of the system than "fault dictionaries"; however, there is a strong system dependence (a new set of rules is needed for every system), and minor changes in the system (upgrades) often render an entire knowledge base obsolete, and the time needed to acquire the knowledge makes it hard to deliver a rule based expert system at the time of delivery of the actual system (it hasn't been operated enough for technicians to figure out how to diagnose it). A more subtle failing in rule based systems is a lack of clarity, a rule saying "IF voltage1=100 and voltage3 = 0 THEN resistor1 = open" gives no clue as to how the components are connected or where these measurements are taken. This approach does not lend itself to maintainability (without a schematic the rule is gibberish). Another failing of rule based systems is that often since there are no requirements or specifications, failures often go unnoticed since there is no idea of what the "correct" behavior should be. Many rule based "FDIR systems" do nothing more than sensor verification; however, once a failure of a sensor is identified they are often incapable of performing any useful diagnosis due to a lack of complete information [19].

The approach which is suggested here would combine elements of the previous approaches, but would use as their backup Model Based Reasoning (MBR). Research in MBR has developed many

mathematically sound methods for doing diagnosis, and has made significant progress in dealing with some of the more difficult fault diagnosis problems: multiple faults, fault masking, unexpected causality (dripping pipes for example or a solder bridge), unanticipated failure modes, and faulty instruments. None of which can typically be dealt with in other FDIR systems. The research in this area has developed many sharply focused techniques [10-19].

The initial efforts in Fault Tolerance Redundancy Management (FT/RM) have used Digraph Matrix Analysis (DMA) to capture connectivity or causality among components and to help develop and verify the design of redundancy management algorithms and fault tolerant systems [35]. DMA can also help in the determination of overall system reliability when used in conjunction with Failure Modes and Effects Analysis Critical Item Lists (FMEA/CIL) work sheets. DMA will expose points of failure that have system wide or safety critical consequences. The DMA tool uses the Warren algorithm [36] for transitive closure to compute reachability. This reachability analysis is typically the first step taken when using a MBR system.

Another tool being used in this effort is the Systems Testability Analyzer (STA) which is part of the Integrated Diagnostic Support System (IDSS) which has been developed in Ada by Harris Corporation for the NAVY. It uses a systems design to determine testability and ambiguity groups, and help improve fault detection probability. The testability analysis flags areas where ambiguity groups arise (you can't figure out which component is faulted), and suggests either further test in operations, or where to put additional instrumentation during design [37]. The Adaptive Diagnostic System (ADS) utilizes an opportunistic approach to doing FDIR. It starts with the simplest model, i.e. compiled, lookup table knowledge, and proceeds through several intermediate stages until it does a full simulation. Two of the intermediate knowledge levels are: analytical and logical. The analytical approach uses dependency models, fault signatures, and BIT information to detect and isolate faults. The empirical approach uses production rules, to help mask BIT/BITE false alarms. ADS can also update its own statistical parameters to reorganize its search more effectively dependent on real world behavior of the system. The algorithm used to generate the decision trees in STA are based on Artificial Intelligence (AI) search techniques and information theory [31]. Ways in which MBR can extend these tools are being investigated.

**Planning**

Another area where automated systems could save a lot of manpower, is in planning and scheduling the resources, manpower, and experiments on the SS. Planning and scheduling has received a lot of attention in the literature [43-49]. This problem is basically intractable in the following ways: job shop scheduling is NP-Hard, creating an optimal schedule is exponentially related to the size of a given problem, determining if an arbitrary plan is feasible is NP-Complete, and determining an admissible first step is NP-Hard [43-44]. The way in which constraints are represented can have a major impact on how well solutions can be determined [45]. A Computer Assisted Scheduling System (COMPASS) has been developed in Ada, using XWindows for the operator interface, which has many features desirable for NASA programs such as: discrete and continous resources, returnable and consumable/producible resources, and can reason about state dependent activities including both boolean and real valued state variables. This system allows interactive, mixed initiative scheduling in several different contexts, multi-level scheduling, multi-interval scheduling, and multi-agent scheduling [46].

**Current Status and Future Plans**

Honeywell continues in the development of the Maintenance Diagnostic System (MDS) which will augment the existing FDIR system in the Attitude Determination and Control System (ADCS) by isolating faults, aiding in preventive maintenance, and maintenance instruction [38]. A port from Sun workstations to PS/2s has also been accomplished. The predictive maintenance aspect takes advantage of empirical relationships, such as the fact that in the lasers in the ISA ring gyros have a fault signature that can be recognized 2-3 months in advance. It turns out that if one plots the lasing power against the input current there is a characteristic curve. However, when a laser starts to go bad its performance strays from this curve in a fairly predictable fashion. It is this sort of trend analysis that KBS are ideally suited for.

As part of MDSSC's participation in the Advanced Automation Methodology Project (AAMP), which is defining engineering methodologies and standards for developing KBS in the SSFP, the recovery

part of an FDIR system for the Space to Ground Communication link is being developed. With the possible movement of the FDIR functions to the ground at Permanently Manned Configuration (PMC) this may be one of the most important links for the space station. This project is using existing tools already developed at Johnson Space Center (and possibly General Electric), and augmenting them by adding the recovery procedure generation function. An IR&D for doing ICAT on the C&T system is being initiated this year.

An effort to enhance the Active Thermal Control System (ATCS) simulation with an external control architecture is being initiated. This is an outgrowth of the Thermal Expert System (TEXSYS) demonstration project. This task order will initially involve MDSSC Thermal personnel in developing the simulation and interfaces, and LMSC in developing the control algorithms, and identifying interesting (not easy to identify) fault modes. This system will be designed with an interface so that an external control system can be implemented (either human, or some automated system).

The DMS contractor (IBM) has been asked to perform a common space station expert system services trade study. This task would involve polling the space station community to provide inputs for a white paper on common expert system services (emphasis on FDIR). They have also been asked to develop a DMS system management FDIR function using KBS techniques.

QMR and DXPlain are two medical diagnosis expert systems which are being considered in a trade study being undertaken by Crew Health Care System (CHeCS) personnel to pick a diagnosis system for the station. A medical diagnosis expert system is important because there will not be a physician immediately available all the time on the ground, and because there may be life threatening conditions when ground support is unavailable (C&T or the Tracking Data Relay Satellite System failed).

### Conclusions

This paper has described a technical approach which should allow the integration of KBS into the SSFP. It is important to remember that KBS are actually "people amplifiers" and should be used to augment both the human and machines capabilities [40]. The current status of several SSFP KBS projects was briefly reviewed. The efficient usage of KBS technology in the SSFP should make the operation of the station much easier and cost effective.

## References


[1] Ferguson, Mary "Machine Intelligence and Robotics at Johnson Space Center" JSC-23518, September 1989.
[2] Erickson, J. D., "Intelligent Systems in NASA Space Missions" to be published: Proceedings international conference on supercomputing in Nuclear Applications, Mito City, Ibaraki, Japan, March 1990.
[3] MDSSC-KSC "KSC Lessons Learned Applicability to Space Station Freedom Support of Exploration Scenarios" Onorbit Assembly/Servicing Task Definition Study.
[4] MDSSC-SSD "Automation and Robotics Plan" MDC-H4115, October 1989
[5] Flaherty, D.R. "Automation and Robotics Plan" (DR-17), MDC H2036A, June 1986
[6] Wood, R.M., McKee, J.W., Kuck, G.A., Flaherty, D.R. "High-Leverage Advanced Development Projects to meet Space Station Requirements" MDC H1479, October 1985
[7] Nii, P. H., "Blackboard Systems" The handbook of Artificial Intelligence Volume IV, Barr, A., Cohen, P. R., and Feigenbaum, E. A. (eds.), Addison Wesley Publishing Co. Inc., New York, NY, 1989.
[8] Klir, G. J., "Architecture of Systems Problem Solving" Plenum Press, New York, NY, 1985.
[9] Smith, K. M., and O'Neil, G., "Application of General Systems Theory for the Definition and design of Advanced Avionics Systems" Tutorial at Digital Avionics Systems Conference, San Jose, California, October 17-20, 1988.
[10] Bobrow, D.G., "Qualitative Reasoning about Physical Systems" MIT Press, 1985.
[11] Davis, R., "Form and Content in Model Based Reasoning" presented at Workshop on Model Based Reasoning AAAI/IJCAI-89, Detroit, MI, August 1989.
[12] Davis, R., and Hamscher, W., "Model Based Reasoning: Troubleshooting" MIT AI Memo No. 1059.
[13] Davis, R., "Robustness and Transparency in Intelligent Systems" Symposium on Human Factors, National Academy of Sciences, Washington, DC, January 29-30, 1987.
[14] Reiter, R., "A Theory of Diagnosis from First Principals" Artifical Intelligence, 32, 57-95, 1987.
[15] Raiman, O. "Order of Magnitude Reasoning" AAAI-86, Philadelphia, Pennsylvania. San Mateo: Morgan Kaufmann Publishers.
[16] Malin, J. T., Basham, B. D., and Harris, R. A, "Use of Qualitative Models in Discrete Event Simulation for Analysis of Malfunctions in Continous Processing Systems" chapter in "Artifical Intelligence in Process Engineering (M. Mavrovouniotis, ed.), Academic Press, 1989.

[17] Forbus, Kenneth "Qualitative Physics: Past Present, and Future" AAAI-88 Tutorial, pp. 239-296.
[18] de Kleer, J., and Williams, B. C., "Diagnosing Multiple Faults" Artifical Intelligence, 32, pp. 97-130, 1987.
[19] Fulton, S. L., and Pepe, C. O. "An introduction to Model-Based Reasoning" AI Expert, pp. 48-55, January 1990.
[20] Bellman, C. L., and Walter, D. O., "Analyzing and Correcting Knowledge Based Systems requires Explicit Models" AAAI-88 Verification and Validation Workshop
[21] Stachowitz, R. A., "Validation of Knowledge Based Systems" Second AIAA/NASA/USAF Symposium on Automation, Robotics, and Advanced Computing for the National Space Program, March 1987.
[22] Nguyen, T. A., Perkins, W. A., Laffey, T.J., and Pecora, D. "Knowledge Base Verification" AI Magazine, 8(2), pp. 65-79, Summer 1987.
[23] Stachowitz, R. A., et. al. "Building validation tools for knowledge based systems" first annual workshop on Space Operations, Automation and Robotics, pp. 207-216, NASA conference Publication 2491, Houston, TX, August 1987.
[24] Culbert, C., Riley, G., and Savely, R.T., "Approaches to the Verification of Rule Based Expert Systems" SOAR 1987
[25] Rushby, J., "Quality Measures and Assurance for AI Software" NASA contractor report 4187, 1988.
[26] Landauer, C., "Principles of RuleBase Correctness" AAAI/IJCAI-89 Workshop on V&V.
[27] Aly, N. A., Aly, A. A. "Measures of Testability for Automatic Diagnostic Systems" IEEE Transactions on Reliability, 37(5), pp. 531-538, December 1988.
[28] Feagin, T. "Real-Time Diagnostic Expert Systems using Bit Strings" To be published.
[29] Pearce, D., "Induction of Fault Diagnosis Systems from Qualitative Models", AAAI-88, pp. 353-359.
[30] Gilmore, J. P., and Gingher, K. "A Survey of Diagnostic Expert Systems" Applications of Artificial Intelligence V, Proceedings of SPIE - The international society for Optical Engineering, Vol, 786, pp. 2-11, 1987.
[31] Pattipati, K. R., Deckert, J. C., and Alexandris, M. G. "Time-Efficient Sequencer of Tests (TEST)" IEEE 1985 Autotestcon proceedings, pp. 49-62, 1985.
[32] Schumacher, H., and Sevcik, K.C. "The synthetic Approach to Decision Table Conversion", Communications of the ACM, 19(6), pp. 343-351, 1976.
[33] Jackson, A. H. "Machine Learning" Expert Systems, 5(2), PP. 132-150, May 1988.
[34] Forgy, C. L., and Shepard, S. J. "Rete: a fast match algorithm" AI Expert, pp. 34-40, January 1987.
[35] Sacks, I. J., "Digraph Matrix Analysis" IEEE Transactions on Reliability, Vol. R-34, No, 5, December 1985.
[36] Warren, H. S., "A Modification of Warshall's Algorithm for the Transitive Closure of Binary Relations" Communications of the ACM, 18(4), April, 1975.
[37] Rosenberg, B. J., "The Navy Integrated Diagnositic Support System - System Overview, Architecture, and Interfaces", Harris Corporation, 6801 Jericho Turnpike, Syosset, New York 11791.
[38] Toms, D., Hadden, G., and Harrington, J. "Attitude Determination and Control System Maintenance Diagnostic System" These proceedings
[39] Narayan, H., and Viswanadham, N., "A Methodology for Knowledge Acquisition and Reasoning in Failure Analysis of Systems" IEEE Systems, Man and Cybernetics, 17(2), pp. 274-288, March/April 1987.
[40] Woods, D.D., "Cognitive Technologies: The Design of Joint Human-Machine Cognitive Systems" The AI Magazine pp. 86-91, 1987.
[41] Clancey, W. J., "Viewing Knowledge Bases as Qualitative Models" IEEE Expert Summer 1989, pp. 9-23.
[42] Clancey, W. J., "Heurestic Classification" Artifical Intelligence, 27, pp. 289-350, 1985.
[43] Fox, B.R., "Minimally Ordered Sets" to appear in Progress in Robotics and Intelligent Systems, C. Y. Ho and G. Zobrist (eds.), Ablex Publishing, Norwood, New Jersey, 1989
[44] Ullman, J.D., "NP-Complete Scheduling Problems" Journal of Computer and System Sciences, 10, pp. 384-393, 1975.
[45] Fox, B.R. "Representational Adequacy in Temporal Reasoning" Submitted to the First International Conference on Principles of Knowledge Representation and Reasoning, Toronto, Canada, May 15-18, 1989.
[46] Fox, B.R. "Mixed Initiative Scheduling" to be published.
[47] Fox, M.S., and Sadeh, N. "Preference Propagation in Temporal/Capacity Constraint Graphs", CMU-CS-88-193.
[48] Allen, J.F. "Maintaining knowledge about temporal intervals" Comm. of the ACM, 26(11), pp. 832-843, 1983.