Computer System Evolution Requirements For Autonomous Checkout of Exploration Vehicles

Second Symposium on the Evolution of Space Station Freedom

August 6 - 8, 1991
League City, Texas

Study Manager:
Mike Sklar
MDSSC-KSC
Mail Code: F530
407-383-7062

Task Manager:
Tom Davis
NASA-KSC
Mail Code: PT-AST
407-867-2780

Agenda

- Overview of Advanced Automation Study
- Overview of Autonomous Diagnostic Checkout Task
- Assumptions
- Analysis Methodology
- Results
- Summary of Requirements for Autonomous Diagnostic Checkout
- Recommendations for Additional Analysis

Overview of Advanced Automation For In-space Processing Study

This study, now in its third year, has had the overall objective and challenge of determining the needed Hooks and Scars in the initial SSF system to assure that On-Orbit assembly and refurbishment of Lunar and Mars spacecraft can be accomplished with the maximum use of automation possible. In this study automation is all encompassing and includes physical tasks such as parts mating, tool operation, and human visual inspection, as well as non-physical tasks such as monitoring and diagnosis, planning and scheduling, and autonomous visual inspection. Potential tasks for automation include both EVA and IVA events. A number of specific techniques and tools have been developed to determine the ideal tasks to be automated, and the resulting timelines, changes in labor requirements and resources required. The Mars/Phobos exploratory mission developed in FY89, and the Lunar Assembly/Refurbishment mission developed in FY90 and depicted in the 90 Day Study as Option 5, have been analyzed in detailed in recent years. The complete methodology and results are presented in FY89 and FY90 final reports.

Overview of Advanced Automation For In-Space Vehicle Processing Study

701

- Study is part of SSF Advanced Studies Program
- Three year study began in FY 1989
- Primary study objectives
  - Identify suitable processing tasks to be automated (physical & non-physical)
  - Determine hooks & scars required to support evolved SSF on-orbit processing
  - Determine impacts of automated processing operations (timelines, reduced labor, SSF resource requirements)
  - Assess required automation technologies

Overview of Autonomous Diagnostic Checkout Task

The Advanced Automation study has struggled with the issue of determining computing resources and the resulting Hooks and Scars required to perform autonomous or semi-autonomous cognitive processing tasks. This issue has been addressed in each of the past three years. However it has been extremely difficult to establish any specific methods or results. The reason for this is the spacecraft to be processed are in a conceptual phase only at this time, and thus the system and processing details are non-existent. This makes it especially difficult to provide details for such high-level tasks such as planning and diagnostics. Furthermore the software environment and architecture for the SSF Data Management System, has not been completely defined yet either.

Thus, in order to provide more specific results this year, the study has begun to focus on a more specific, less encompassing task. The problem of complete system checkout and diagnostics of a vehicle after it has been readied for launch is an ideal task for nearly complete automation. Because this task will be similar for both exploration and existing vehicles, such as the Space Shuttle, detailed information can be obtained. Thus, computer requirements for complete on-orbit checkout of a vehicle, assuming a single IVA astronaut monitoring the testing, have been determined. These requirements assume a baseline computer system is available either on board SSF, on board the vehicle or possibly on the ground. Thus, only those requirements which are specific to autonomous checkout are determined.

Overview of Autonomous Diagnostic Checkout Task

- Identify the vehicle processing tasks that can benefit from AI technologies

703

- Develop a methodology primarily focused on determining the additional computer system memory requirements for autonomous diagnostic checkout

- Estimate the additional computer system requirements for on-orbit diagnostic checkout of exploration vehicles

- Provide input to DMS evolution requirements and studies

Assumptions

A number of assumptions have been made to confine this task and make possible the determination of specific numerical requirements. Most of these are straightforward. Note that task times for both the case of human test conductors performing each step and autonomous checkout are identical for a majority of system tests. This is true because the total checkout task time consists mostly of physical processes and measurements. That is the time to issue a system command or diagnose a problem is usually much less than waiting for a tank to fill or a sensor reading to settle etc.

Assumptions

- Focus on autonomous diagnostic checkout of LTV at SSF
- Computer memory forecasts are in addition to baseline DMS requirements and are independent of where the expert systems reside (ie. on SSF, on LTV, or on the ground)
- All test support equipment configuration and hookups are completed prior to conducting any automated diagnostic checkout procedures
- The memory forecasts do not include additional requirements for diagnosing problems with support equipment
- Automated diagnostic checkout test will take same elapsed time as manual diagnostic test, and only one crew member is required to supervise the automated diagnostic checkout expert systems
- All detailed test analyses based on expert system technology

Analysis Methodology

A detailed method consisting of various levels of detailed analysis, extrapolations and generic guidelines has been developed to analyze and determine automated diagnostic testing requirements . This can be applied to any proposed system which can be modeled as a set of analogous currently built systems for which well documented checkout procedures exist. The requirements are based on the use of expert systems technology being used to perform the automated diagnostic procedures. The basic results provide the number of logic rules and object data facts necessary to perform the required test. This information is then used to predict computer resource requirements such as processor and mass storage memory.

The method consists of 4 specific tasks or components to provide the final requirement data:

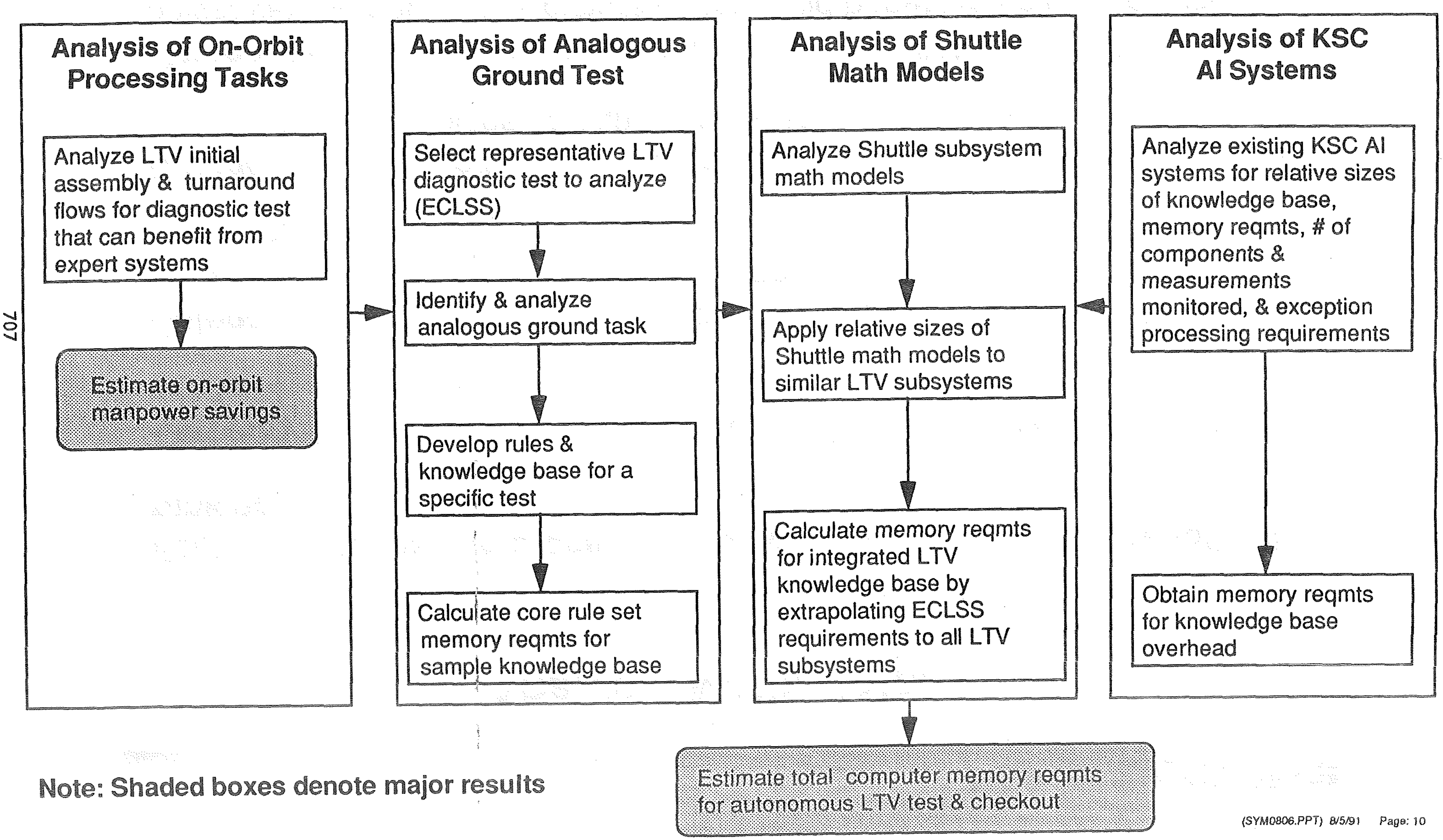
1. First the processing tasks of an entire system assembly or refurbishment process are analyzed to determine which tasks are purely diagnostic tasks that are beneficial to be automated. Because a single IVA based operator can completely monitor the checkout test the required on-orbit labor is reduced.

2. One or more analogous, currently existing systems are then identified. The analogous ground task(s) is then analyzed and a complete set of rules and full knowledge base for the one system or test item is developed. From this example coded knowledge base the memory requirements per object or specific test item are determined.

3. The relative sizes of all vehicle systems are then determined by examining the relative sizes of all analogous systems. The relative sizes of shuttle systems is determined from the relative sizes predicted by STS math model sizes. The math models are used to simulate system performance for training and accurately represent total number of components and complexity in each system.

4. Other existing AI systems currently in field use are then analyzed to provide additional missing data items. Because these systems actually exist, the size and complexity of the physical systems they operate on are well known. Also, because they are in use, the overall computer requirements are well known. Thus they provide, in a sense, a reality check to the predicted requirements determined for on-orbit checkout. These existing ground systems can also be used to estimate factors such as graphics display and user interface requirements, operating systems and other factors which represent requirements in addition to those requirements due to specific rules and knowledge.

Analysis Methodology



Note: Shaded boxes denote major results

Analysis Methodology
Analyze On-Orbit Processing Tasks

- **Each LTV processing procedure/task was assigned one of the following rankings**

  Category 1 - A procedure/task that is a physical task done by an astronaut or telerobot

  Category 2 - A procedure/task that could benefit from advanced AI technology (ie. vision systems, pattern matching, inspection) beyond the scope of this expert system analysis

  Category 3 - A procedure/task that is strictly a diagnostic test and/or checkout that can be accomplished by an expert system

  Category 4 - A procedure/task that is primarily an IVA astronaut activity (ie. power vehicle down, take pictures) that could benefit marginally by using AI/expert systems

- **An estimate of manpower savings for all Category 3 tasks was calculated as a result of utilizing only one IVA astronaut to supervise expert system vs. baseline 2 - 3 astronauts per checkout task**

Analysis Methodology
Analyze an Analogous Ground Task

- Analyze the diagnostic checkout procedures for the Atmospheric Revitalization and Pressurization Control System (ARPCS) of the Environmental Control and Life Support System (ECLSS)

- Interview the Shuttle ECLSS system engineers on standard diagnostic checkout procedures and exception processing procedures (troubleshooting)

- A sample knowledge base was built for an ARPCS cabin pressure relief valve test as documented in OMI (Operations & Maintenance Instructions) V1020

Analysis Methodology

Why Analyze the ECLSS and ARPCS?

- The ECLSS is a good representative vehicle system (303 fluid components and 330 instrumentation measurements)

- The ARPCS is a complex subsystem (22.5% of the measurements and components in ECLSS) for which a sample knowledge base could be encoded

- Forecasts of memory requirements could be calculated for ARPCS and ECLSS as a result of building the sample knowledge base

- Memory estimates obtained for ARPCS and ECLSS could be extrapolated to all vehicle systems

- Provides a real-world example of diagnostic checkout procedures on a space vehicle

Analysis Methodology
Analyze Current AI Systems at KSC

- **Knowledge-Based Autonomous Test Engineer (KATE)**
  KATE is an excellent example of an AI system that monitors gauges, valves, flow rates, and pressures. It was observed while monitoring the LOX tanking process of the External Tank during the launch countdown for the STS-40 mission.

711

- **Operations Analyst Expert System (OPERA)**
  OPERA is an intelligent operator assistant that monitors Firing Room hardware. It reacts to problem situations and notifies the Master Console Operator when hardware failures are detected.

- **Expert Mission Planning and Replanning Scheduling System (EMPRESS)**
  Expert system that schedules Shuttle resources at KSC. A significant portion of the knowledge base deals with handling exception processing.

- **Each system provided data on the relative sizes of the knowledge bases, memory requirements, disk storage requirements, and the number of components and measurements monitored.**

Analysis Methodology
Analyze Math Models of Shuttle Systems

- Shuttle subsystem math models are used for training and certification of Firing Room console operators

- They are based on the number of components and measurements in each vehicle system, and therefore reflect the relative size and complexity of each system

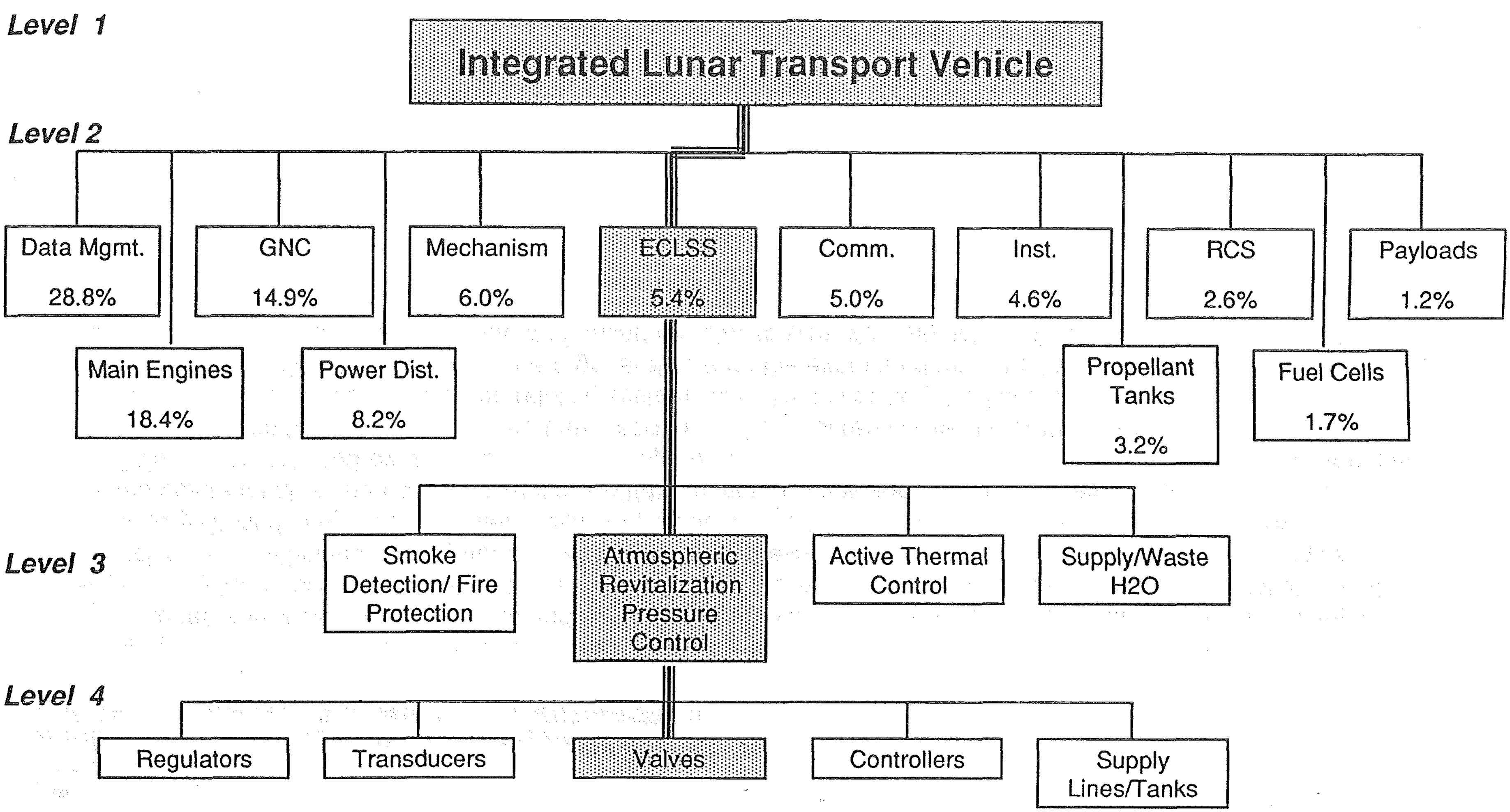
- A repartitioning of the current Shuttle vehicle systems into projected vehicle systems of the future LTV was conducted to accommodate for differences between the two vehicles

- The relative size of each vehicle system was needed to extrapolate the computer memory requirements for all vehicle systems once the ECLSS estimate was obtained

Approach For Estimating System Requirements

Although analogous systems in use today exist for every system in an exploration vehicle, and thus specific checkout requirements exist, due to the size and number of systems, it is far too laborious to actually develop sample knowledge bases for each system. Instead one specific, highly representative task has been analyzed in detail. In this case a specific valve within the Atmospheric Revitalization and Pressure Control System (ARPCS), a subsystem of the Environmentally Closed Life Support System (ECLSS), on board the Shuttle was analyzed in detail. Based on documented Operational Maintenance Instructions (OMI), an actual knowledge base for a valve checkout task was created. This is accomplished by developing a PC computer based expert system using the ECLIPSE expert system shell tool. The requirements for this single test are then used to predict total requirements for the ARPCS which are then extrapolated to the entire ECLSS based on relative system size. That is basic guidelines showing number of rules and memory required per object and system test are generated from the example data. The total requirements for each of the other vehicle systems are then simply computed based on relative size with respect to ECLSS.

Approach for Estimating Expert System Requirements For LTV Diagnostic Test & Checkout



Results Of Building Sample Knowledge Base

A single valve test, the Cabin Pressure Relief Valve, within the ARPCS subsystem of the Shuttle ECLSS system was coded into an actual expert system knowledge base on a PC. The nominal test, in which no unexpected results or problems occur, was based on an actual Operation and Maintenance Instruction (OMI) in use today. The results provide guidelines and expressions to predict the total memory requirements of a knowledge base of any system whose number of components and required measurements are known. Results show that each object in the system requires three facts to represent its state. An object may be a component or a measurement. Facts are simply data items about an object such as its state, (ie. open, closed, on etc.) or a minimum or maximum value for a measurement for example. Rules define either the condition of the system based on the facts or they control the flow of the test. Any test requires a standard set of steps such as start systems, read data items etc. Each test then has specific rules which make up the remainder of the flow for a nominal test. Memory is also required to access current and historical sets of specific fact data items stored in a database. Memory is also required to store the actual expert system inference engine which sequences or applies the rules for a given knowledge base.

Results of Building a Sample Knowledge Base For Cabin Pressure Relief Valve Test

- Results of this exercise showed that there are approximately:
  - **3 facts per object** (an object is defined as a measurement or component)
  - **40 bytes per fact** (a fact is for example "relief valve 1 is open")
  - **270 bytes per rule**
  - **15 Generic Baseline Rules required per test sequence**
  - **Test Specific Rules required per test = 20% times the number of objects**
  - **70K of memory required for database access**
  - **282K of memory required for an inference engine**

Results Of Exception Processing Analysis

The OMI procedures, and the example knowledge base developed based on these, do not include the knowledge and steps taken when failures occur during testing. To accurately predict realistic requirements for a complete checkout system the ability to handle exceptions from the nominal test case must be accounted for. In order to analyze this portion of the system the exception handling rule requirements were determined by examining the ARPCS system as a whole. Interviews with NASA test conductors for this test were carried out to obtain these results. They provided an indication of which systems are most likely to fail and how often exceptions occur during typical testing. For problems which occur frequently, specific test sequence flows developed by the conductors were used to predict exception requirements. It should be noted that this is in essence expert knowledge and experience. Because the exceptions are handled on a case by case basis no documentation for these flows exists. This analysis provides data in the same form as the Relief Valve test but is given in total for the ARPCS system. The total numbers for exception handling in the entire ECLSS checkout are then extrapolated based on the relative size of ARPCS within ECLSS.

Results of Exception Processing Analysis in ARPCS & ECLSS

- ARPCS was analyzed to determine the procedures/tasks that are carried out when a problem is encountered while conducting a diagnostic test
- Procedures and knowledge obtained from expert interviews with NASA test engineers
- Results of this analysis showed that within the troubleshooting procedures for ARPCS there are approximately :
    - 79 objects
    - 237 facts
    - 120 Generic Baseline Rules
    - 656 Test Specific Rules

Results of Exception Processing Analysis in ARPCS & ECLSS (cont.)

- Since ARPCS comprises 22.43% of ECLSS, the exception processing requirements were extrapolated for the entire ECLSS
- Results of this extrapolation indicate that within the troubleshooting procedures for ECLSS there are approximately:
  - 352 objects
  - 1,056 facts
  - 535 Generic Baseline Rules
  - 2,925 Test Specific Rules

Results of Analyzing Current AI Systems at KSC

- Analyzing OPERA revealed that the application specific graphical user interface (GUI) takes up approximately 30% of the total knowledge base requirement

- The analysis of EMPRESS showed that 73% of the knowledge base consisted of rules to handle exception processing

- While analyzing KATE, it was learned that approximately 400MB of disk storage is required by the Shuttle Launch Processing System (LPS) to store 4 - 6 hours of real-time vehicle data

Breakdown of Memory Requirements For ECLSS Knowledge Base

Factor	Memory	How Obtained
Core Object Memory	76.0K	(633 components in ECLSS) * (3 facts per object) * (40 bytes per fact)
Generic Baseline Rule Memory	4.1K	(15 Generic Baseline Rules) * (270 bytes per rule)
Test Specific Rule Memory	34.2K	(633 objects) * (20%) * (270 bytes per rule)
Exception Proc. Rule Memory	980.0K	(219.8K memory required for exception processing in ARPCS) / (22.43% - which is the relative size of ARPCS to ECLSS)
Database Access	70.0K	(Estimated from prior experience in building expert systems)
Application Specific GUI Memory	350.0K	(76.0K + 4.1K + 34.2K + 980.0K + 70.0K) * (30% for application specific GUI)
Total For Knowledge Base	1.51 MB	

Computer Memory Requirements For ECLSS

1.51 MB for ECLSS knowledge base

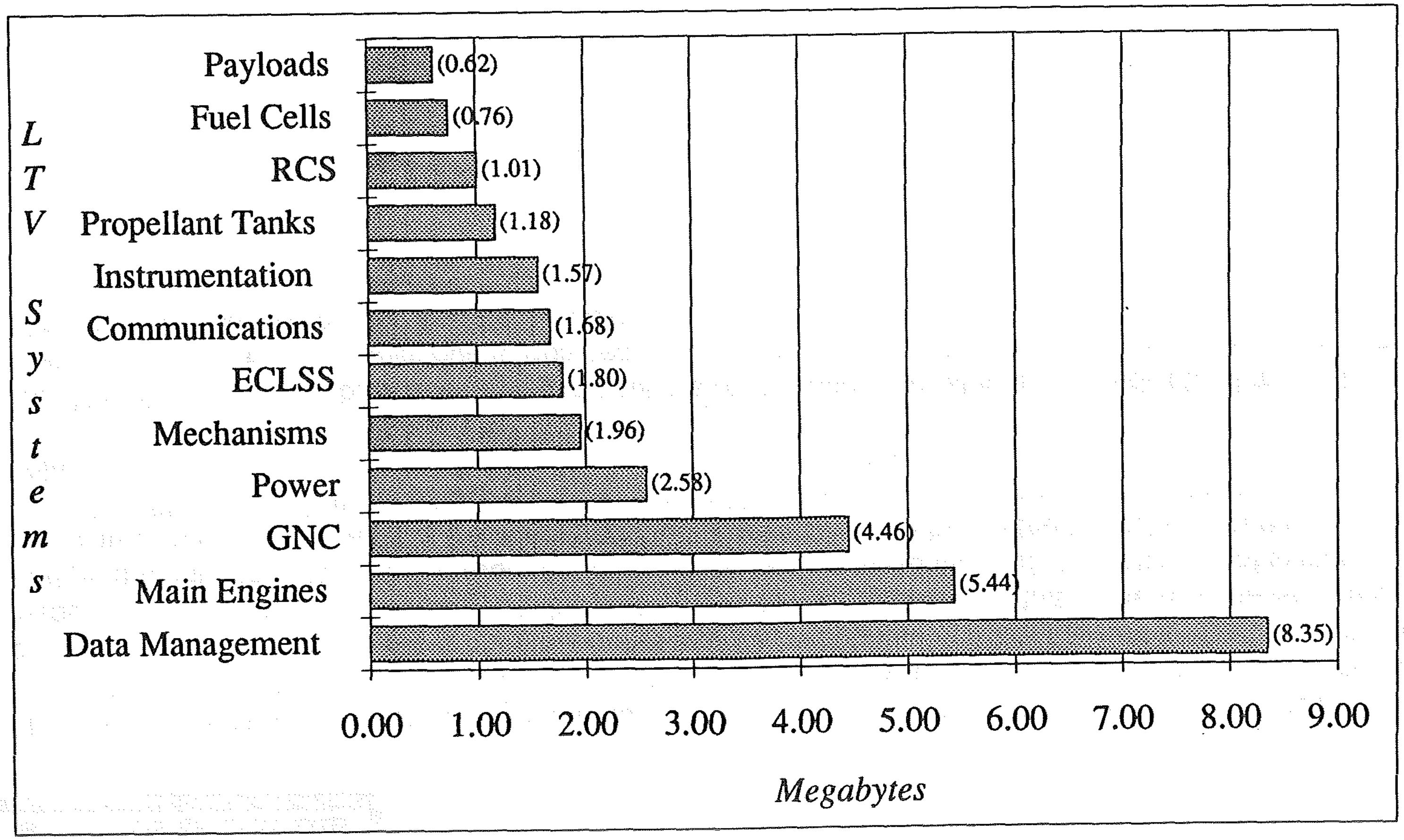
plus

.28 MB for inference engine

equals

1.80 MB memory required to run ECLSS expert system

Graph Illustrating Memory Requirements When Running A Single Expert System At Any One Time



Summary Of Requirements

The primary result of this analysis is the memory required per LTV system to represent autonomous checkout knowledge. The results have been obtained by examining the ECLSS system in detail and extrapolating the results, based on relative system sizes (number of components and complexity), to the other systems. If all systems are run simultaneously a total of 30MB of memory would be required. However, although this is the way the Shuttle is monitored and tested prior to launch, it may not be required for On-Orbit vehicle systems. Some partitioning of system tests may be allowable. This reduce the overall requirement to a number somewhere between the 8.4MB required for the minimum system and the 30MB total. Further analysis will be required to determine what if any test partitioning would be acceptable.

The IVA labor savings is based on previous analysis by this and other studies to predict total manual checkout labor requirements. The savings are totally due to only requiring one supervisor when using an autonomous system as opposed to three test conductors for the manual case.

Summary of Requirements for Autonomous Diagnostic Checkout

- 8.35 MB memory required to execute largest LTV expert system (Data Management)
- Approximately 30MB of memory required to execute all LTV expert systems simultaneously
- 400 MB disk space required for recording 4 to 6 hours of real time vehicle data
- 54% IVA manpower savings for LTV test/checkout using automated diagnostic checkout procedures with single astronaut supervising (160 man-hours for initial assembly and 816 man-hours for refurbishment saved)

Recommendations For Additional Analysis

Two primary extensions of this analysis to provide more detailed results and justification for computer requirements are possible. First, the guidelines used to predict total requirements for each system based on its number of components is based on a single sub-system test example knowledge base. This analysis could be extended by developing example knowledge bases for other systems. The assumption that expert system technology would be used to perform autonomous diagnostics may also be affecting the results. There are a number of other emerging artificial intelligence technologies which may provide significant advantages and a different set of computer resource requirements. For convenience the technologies which could perform diagnostics are defined below:

Expert System - an intelligent computer system that uses knowledge in the form of rules to solve problems that are difficult enough to require significant human expertise for their solution.

Model-Based System - a computer system that takes knowledge about the components of a particular system and applies search and algorithmic techniques to evaluate the performance between the model and real system.

Neural Network - a computer system that can be "trained" to classify information and matches the functionality of the human biological decision making process in a very fundamental manner.

Fuzzy Logic - Systems which are extension of expert systems which involve degrees of probability applied to specific rules and conclusions. They are better at handling real world occurrences which are approximate such as marginally working, working well but not perfect etc.

The overall impacts of using various technologies, or even how to compare requirements of these systems in a general way is not known at this time. Furthermore the results provided here are only for system diagnostics. The same techniques or similar analyses must be done to determine computer requirements for all advanced automation tasks.

Recommendations for Additional Analysis

- Compare and/or contrast the use of rule-based systems, with new and/or different AI technologies. For example; What would be the impact of using model-based systems versus rule-based systems?; Are neural networks applicable?

- Conduct a detailed analysis of the number of measurements and components in other vehicle systems (ie. power distribution, fuel cells, reaction control systems, etc.) to refine the relative sizes of the LTV systems.

- Develop additional sample knowledge bases to validate memory factors

- Develop optimal, acceptable partitioning of system checkout tests to be run an a sequential manner to reduce overall requirements

- Perform expert system analysis for test support equipment (GSE fails more frequently than flight hardware)