The HAL 9000 Space Operating System Real-Time Planning Engine Design and Operations Requirements

Howard K Stetson[1] and Dr. Michael D Watson[2], Ph.D
Marshall Space Flight Center, Huntsville, Alabama, 3581, USA

Ray Shaughnessy[3]
Marshall Space Flight Center, Huntsville, Alabama, 35812, USA

In support of future deep space manned missions, an autonomous/automated vehicle, providing crew autonomy and an autonomous response planning system, will be required due to the light time delays in communication. Vehicle capabilities as a whole must provide for tactical response to vehicle system failures and space environmental effects induced failures, for risk mitigation of permanent loss of communication with Earth, and for assured crew return capabilities. The complexity of human rated space systems and the limited crew sizes and crew skills mix drive the need for a robust autonomous capability on-board the vehicle. The HAL 9000 Space Operating System[2] designed for such missions and space craft includes the first distributed real-time planning / re-planning system. This paper will detail the software architecture of the multiple planning engine system, and the interface design for plan changes, approval and implementation that is performed autonomously. Operations scenarios will be defined for analysis of the planning engines operations and its requirements for nominal / off nominal activities. An assessment of the distributed real-time re-planning system, in the defined operations environment, will be provided as well as findings as it pertains to the vehicle, crew, and mission control requirements needed for implementation.

I. Introduction

The HAL 9000 Real Time Planning / Re-Planning system is a series of distributed software systems that essentially divides planning into its specific functions with collaboration to achieve an overall plan to be executed in real-time. The division of specific planning functions (Power, ECLSS, Communications, etc.) allows a modular design to be applied to support various targets to be operated. Continuous monitoring of the real-time plan execution provides an automatic response to changes or anomalies during the target systems operation. The collaboration and internal approval mechanism for validating a new plan to be executed occurs in much the same manner as teams of human planners perform today. The overall system thus provides autonomous operation to any system that can be commanded in real-time. Assessment of the planning system is driven by operational scenarios, Design Reference Missions (DRM's) for determination of external tools and applications that may be needed to assist in planning and plan verification. The DRM's presented here represent the initial most difficult scenario's to assess.

II. HAL 9000 Planning Executive Design

The HAL 9000 System design employs 9 planning executives and each executive consists of 17 executables, 15 of which are in continuous execution as shown in Figure 1. The Initialization executable performs the initial startup of the planning engine, determines the overall configuration (number of planning engines, their type, intelligence levels, etc) and then starts the Mission Definition File (MDF) Translation Executable. The MDF Translation executable translates the Mission Definition File, which provides the initial plan of automated activities for the specific sub-system. This builds the initial linked list that represents the sub-systems plan. The Safety and MAIN planning Engines build a complete linked list structure of all automated activities (complete plan). The Monitor Activity Parameter Update executable monitors all the specific subsystem autonomous activities' current status and provides execution control parameter updates in real-time. Monitor Logic Modules are added here by the users for the logic utilized in making the parameter updates. The Monitor Time Update executable monitors all the specific

[1] Computer Scientist, Engineering Operations, MSFC/MS166
[2] Discipline Lead Engineer for Operations, Space Launch System, MSFC/MS166
[3] Technical Assistant, Mission Operations Laboratory, MSFC/MS166

subsystem autonomous activities current status and provides execution Time updates in real-time. Monitor Logic Modules are added here by the users for the logic utilized in making the Time updates. Time updates pertain to modifying an activities execution time. The Monitor Stop Warning executable monitors all the specific sub-system

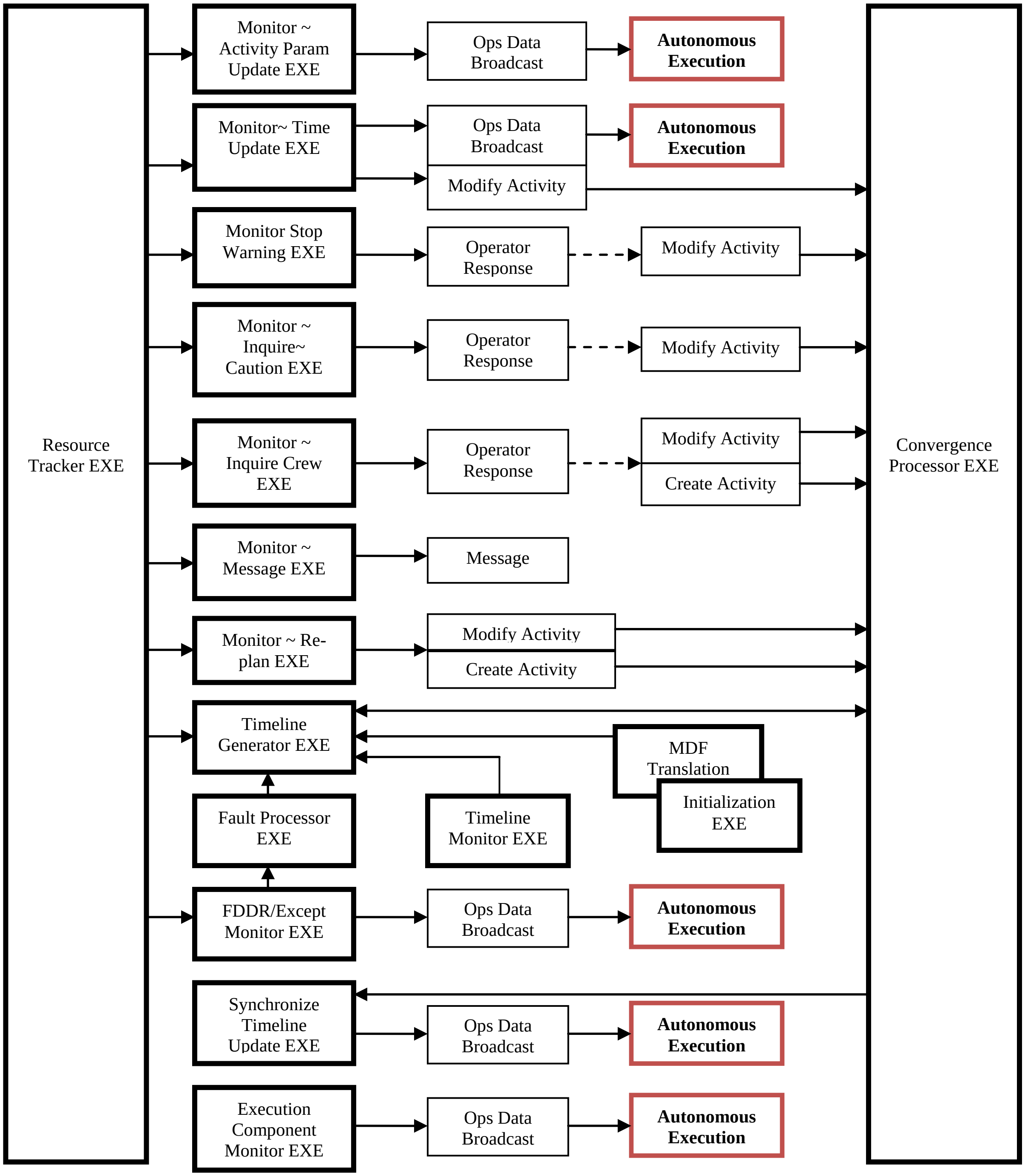


Figure 1. HAL 9000 Executive Software Context

autonomous activities current status and provides a "Stop Execution" of an activity when Warning conditions are met as specified in the Monitor Logic Module added by the user. The Monitor Inquire Caution executable monitors all the specific sub-system autonomous activities current status and provides a Caution Message and inquires the crew for an activity when Caution conditions are met as specified in the Monitor Logic Modules. The Monitor Inquire Crew executable monitors all the specific sub-system autonomous activities' current status and provides a Message and inquires the crew for input into an activity when required conditions are met as specified in the Monitor Logic Modules. The Monitor Message executable monitors all the specific sub-system autonomous activities' current status and provides a Message of information to the crew on an activity when required conditions are met as specified in the Monitor Logic Module. The Monitor Re-Plan executable monitors all the specific sub-system autonomous activities' to determine if a re-plan of an activity or activities is needed as specified in the Monitor Logic Modules. The Timeline Generator executable is started by the MDF Translator Executable to generate a time ordered array of activities for input into the Convergence Processing task for approval by all other planning engines in the system. This is also started by the fault processor when faults encountered during activity execution create the need for a new plan. This task determines whether a re-plan is needed or not after auto-recovery of the fault. The FDDR Exception executable monitors the specific sub-systems hardware and software for failures and notifies the Fault Processor for any re-plan if needed. The Fault Processor executable determines if a new plan is required when a subsystem hardware or software fault is detected by the FDDR Exception Monitor executable and notifies the Timeline Generator if a new plan is needed. Most hardware/software faults are handled by the Execution Component autonomously, then it requires determination of whether the fault effected a change in the current plan. The Synchronize Timeline Update executable creates a new linked list of activities upon successful plan convergence and approval from all other planning engines and coordinates the change over from the old plan to the new plan to insure all planning engines are in sync. The Execution Component Monitor executable monitors the corresponding Execution Component for off nominal status. It can restart sequences that have terminated abnormally or install/remove automated procedures specific to the sub-system. The Resource Tracker executable tracks all the specific sub-system resources for availability during real-time planning re-planning and utilization during real-time execution. The Convergence Processor executable coordinates the subsystem plan approval process with all other planning executives where successful convergence results in a newly approved plan by all sub-systems. The Timeline Monitor executable monitors the specific subsystems activity timeline for adherence as some autonomous activities may over utilize resources and execution time and it also catches anomalies within the Monitor Logic Modules in the Monitor Re-Plan executable.

III. Monitor Logic Modules

The Monitor Logic Modules (Figure 2), cyclically compare the real-time Executive's sub-system status with the sub-system's current activity planned status and the current utilizations. Currently defined modules are: Control Parameter Update logic, Sub-system Re-Plan logic, Time Update logic, Stop-Execution and Crew Inquire logic and simply a Crew Inquire logic capability. Control Parameter Updates are for controlling capabilities within an activities execution, such as a drilling speed, desired depth or length of drilling time as examples. Re-Plan logic monitors are used for detecting conditions within an activity or sub-system that will always require a Re-Plan due to time or resource conflicts. Time Update monitors are used for detecting time constraints for a sub-system or activity and can adjust the execution run time within the allowable slip time of an activity. The Stop Execution and Enquiry Crew logic modules are used for detecting conditions that may not have an auto-response programmed and requires crew input to determine further actions such as a Re-Plan. The Messaging Only logic is used for informative messaging to the crew at any time. Monitor Logic Modules are plug and play code shells developed by the user (crew while on orbit), specific to the subsystem planning requirements and the role the Executive is assigned.

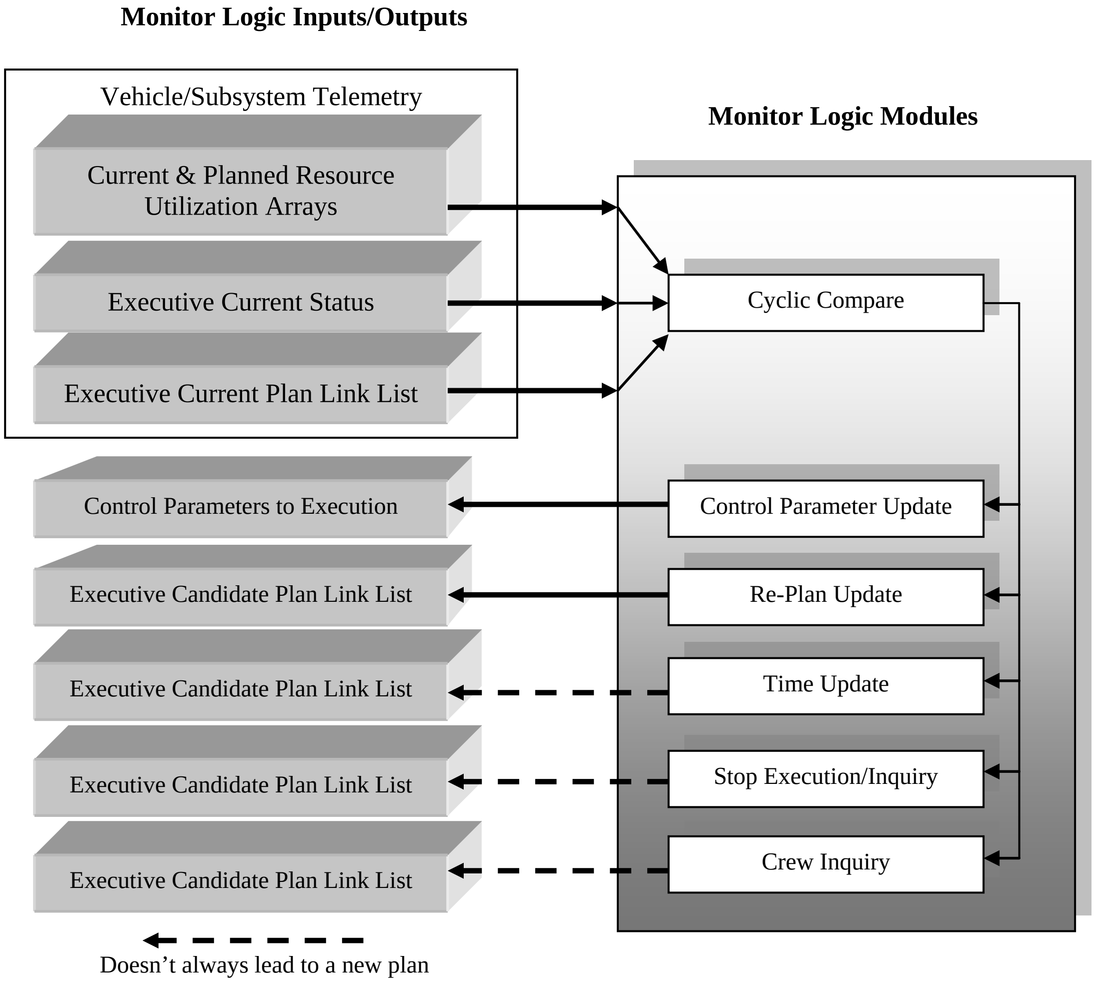

Monitor Logic Inputs/Outputs
Vehicle/Subsystem Telemetry
Current & Planned Resource Utilization Arrays
Executive Current Status
Executive Current Plan Link List
Control Parameters to Execution
Executive Candidate Plan Link List
Executive Candidate Plan Link List
Executive Candidate Plan Link List
Executive Candidate Plan Link List
Doesn't always lead to a new plan
Monitor Logic Modules
Cyclic Compare
Control Parameter Update
Re-Plan Update
Time Update
Stop Execution/Inquiry
Crew Inquiry

Figure 2. Monitor Logic Modules

IV. Timeline Generator Control

Activity Priority is the key to successful plan generation and real-time scheduling and updates. The HAL 9000 Executives utilize a priority range from 100.0 (highest) to 0.01 (Lowest). An overlap in priorities can be provided so that one particular system does not have complete priority over another. During timeline generation, all activities are attempted to be scheduled regardless of priorities. Priorities are utilized during conflict resolution. The priority granularity allows for up to 10000 different priorities and provides a vast distinction between activities. In addition, priorities can also be negative for immediate allowance and automatic inclusion into the plan. For example, a toaster which has negligible power consumption, can be assigned an “anytime” priority for unplanned actions. When the crew powers the toaster, the power monitoring sequence notifies the power executive about the un-planned power up of the toaster, the executive adds this as part of the plan when it encounter’s the toaster’s “anytime” priority. The new candidate plan is exchanged with all other executives for convergence, and the toaster power up just became part of the new plan. The “anytime” feature to planning events and activities is included to provide the crew the level of autonomy needed for day to day living, without having to plan everything for every minute. Priority Leveling is the second control for adjusting a plan. Priority Leveling is applied during Timeline Generation and conflict resolution. Priority Leveling ranges from .1 to 10.0. Priority Leveling of .1 provides a difference between activity priorities by .1 and a level of 1.0 allows all activities with a priority within a range of 1.0 to be scheduled as the same priority. For example, an activity with a priority of 60.2 and an activity with a priority of 60.9 would be scheduled as if they had the same priority. The crew can set the Priority Leveling via the Executives and the default Priority Level is set initially upon Mission Definition File input.

V. Convergence Processing

Convergence Processing (figure 3) is where each sub-system submits its plan to the other subsystems for identification of conflicts and verification. Return codes are passed back to the sending executive denoting a pass or fail condition and the reason why (Reason Codes and Data). Plan updates are made autonomously and submitted back for another round of "convergence" until a pass condition is returned by all Executives (the plan has been approved by all) or until the maximum number of convergence attempts have been made, which at that time, the crew is inquired. The current conflict resolution reason codes are as follows:

Conflict Resolution (Reason Codes)	**Resultant Actions**
Resource constraint	(Resource Utilization Arrays, Resource ID)
Not Available	(Proposes new time based upon availability, slip-time)
Concurrent Use	(Schedules highest priority, submits new time, Inquire Crew)
Limit Violation	(Proposes new time, if new time not found then Inquire Crew)
Rule Constraint	(Proposes new time, Rejects activity input, Inquire Crew)
Time Constraint	(Proposes new time)

The convergence limit is the number of attempts a plan is exchanged between Executives for validation and approval. If one Executive does not accept an edit to the timeline, the edit is modified based upon the reason codes received and re-exchanged. If more than one Executive rejects an edit to the timeline, each reason code is examined and adjustments are made for each before the next exchange is initiated. Once the Convergence Limit is reached with no validation and approval of the timeline, the crew is called in for disposition of the timeline. An un-converged timeline can be approved for execution by the crew, effectively an over-ride to execute, or the crew can make a manual edit based upon the rejection reason codes and re-submit the timeline to the Executives for another round of convergence processing. Once validation and approval has been successful, the complete timeline is re-scanned by the Safety Executive for rule violations, and if none, each Executive broadcasts its timeline to their Execution Component. Upon verification of receipt of the new timelines from the Execution Components, the Main Executive requests all Executives and Execution Components to "work" the new plan. This process is called a "Synchronized Timeline Operations Update". The old plan is discarded, memory is freed, and the new plan becomes the current plan for each subsystem.

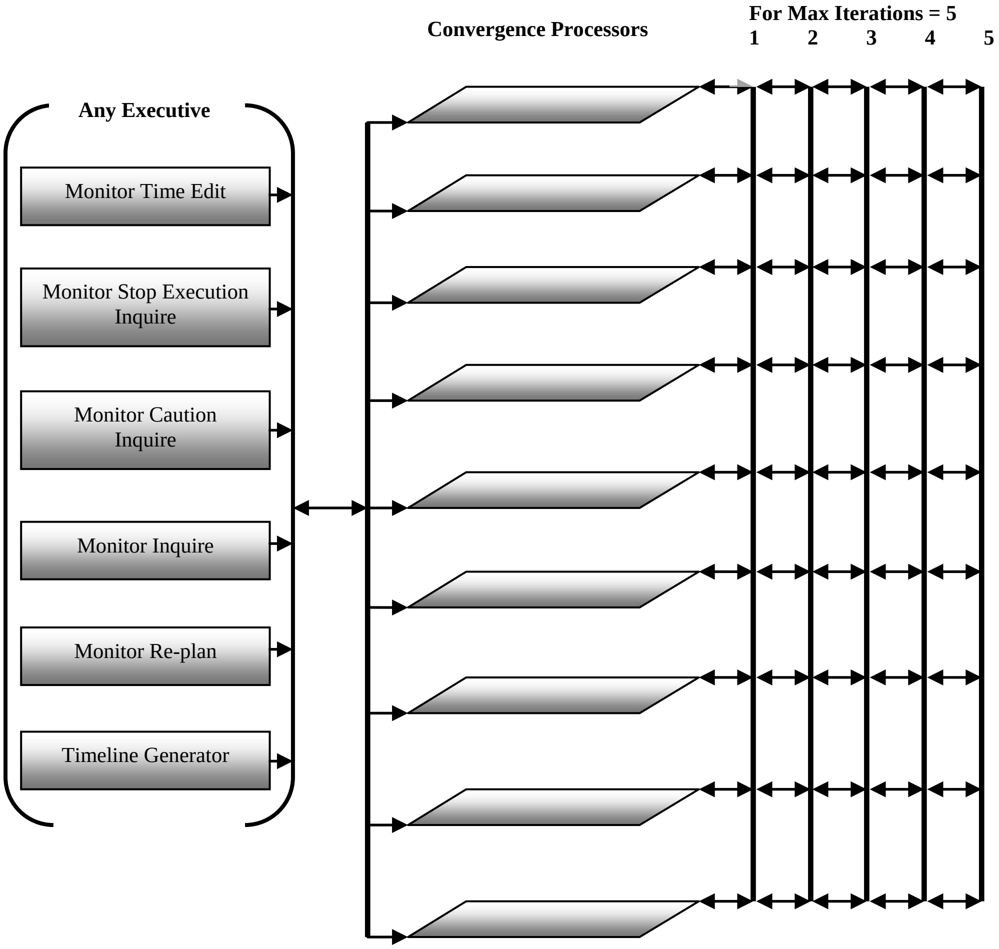


Figure 3. Convergence Processing

VI. Memory Structures

The HAL 9000 Executives build link list structures of their targeted systems/sub-systems (Figure 5) and also the current activity plan (Figure 6). The Executive Linked List Structure is created upon startup of the specific Executive, where Knowledge Pack data is read to determine the structure of the system to be planned by system, sub-system and the devices attached. This is the core structure used by an Executive for knowledge about its system. The Activity Linked List is the core structure used by an Executive for knowledge about its plan within the system. The Main and Safety Executive use a linked list structure of the complete plan of all systems. The Activity Linked List structure is used to create a time ordered array of activity entries for consumption at the Execution Component. Each activity entry in the array contains the time and state conditions for the system, sub-system, or devices. The Execution component utilizes the time order array for tracking current execution states as it pertains to the plan of state changes within the system, and optionally, commands the state changes that are resident within the time ordered array, this latter option is called Full Auto Mode. Full Auto Mode can be set at the specific Executive or alternately at the corresponding Execution Component.

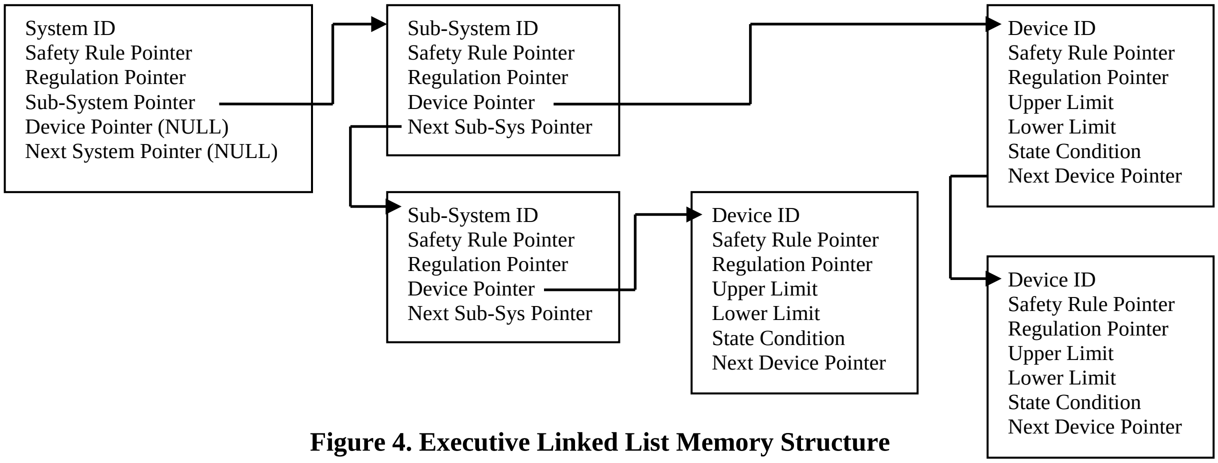


Figure 4. Executive Linked List Memory Structure

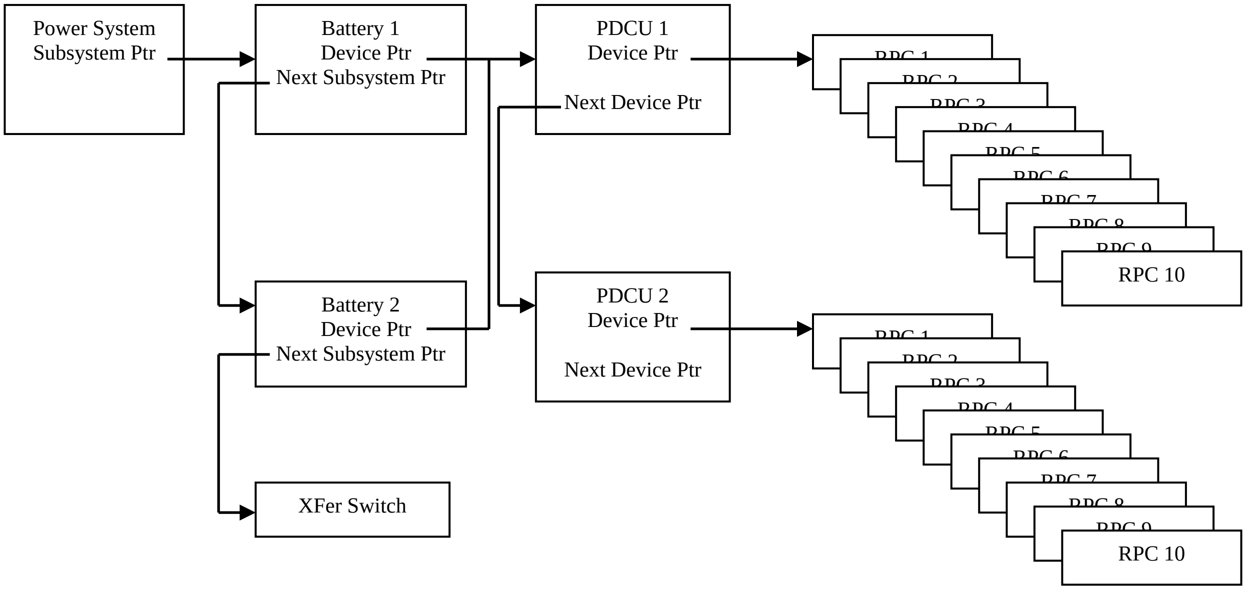


Figure 5. Example Executive Linked List Structure

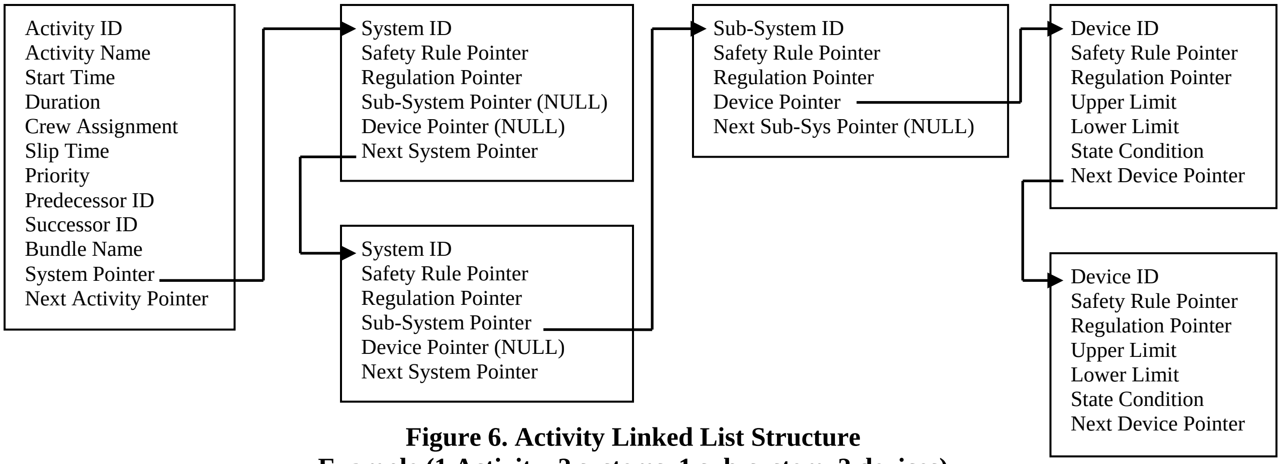


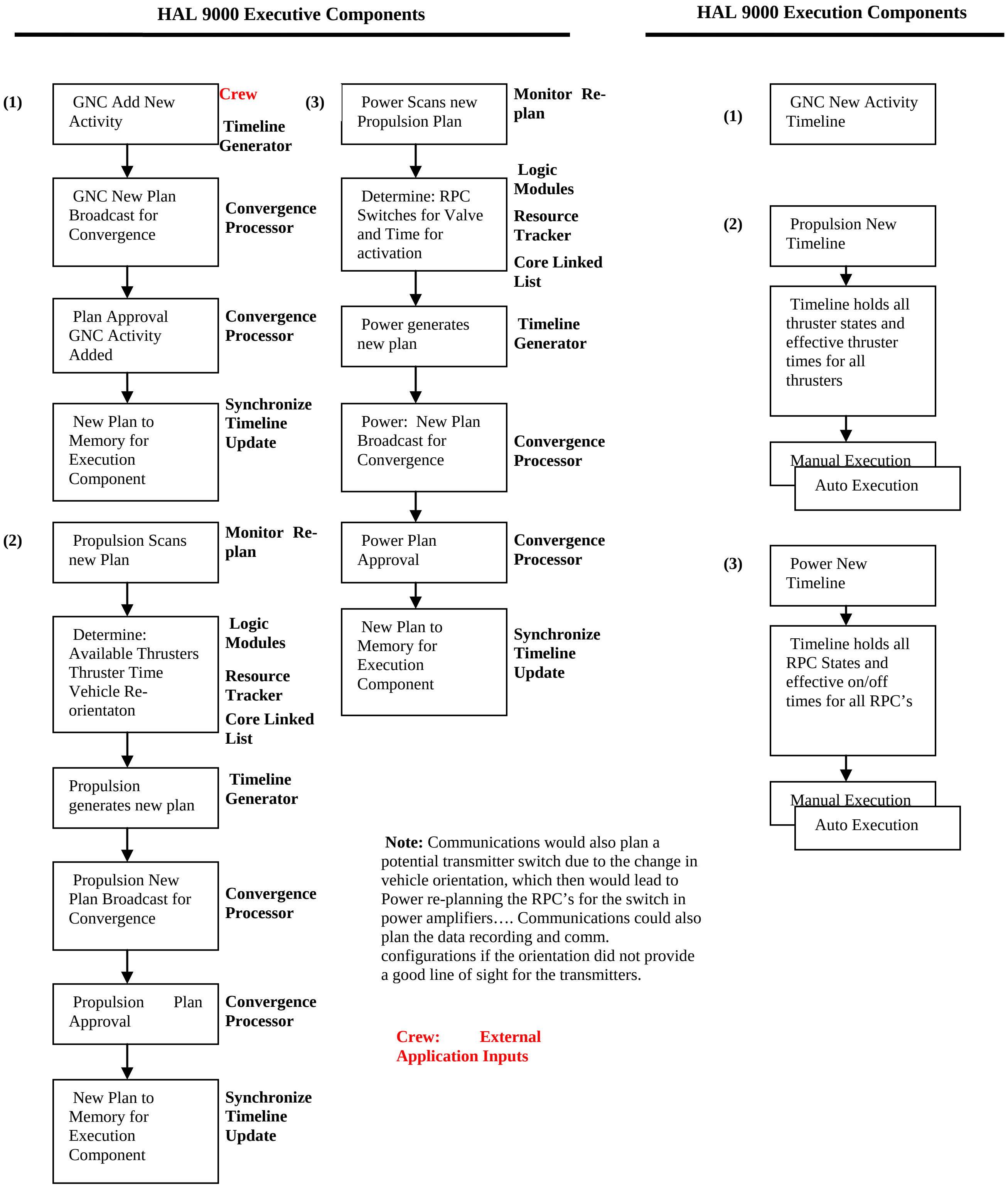
Figure 6. Activity Linked List Structure
Example (1 Activity, 2 systems, 1 sub-system, 2 devices)

VII. Design Reference Mission Introduction

The moons of Jupiter represent an intriguing destination as Human Exploration of the Solar System expands beyond the inner asteroid belt planets. Europa and Ganymede are perhaps the most interesting candidates for initial exploration. The path to Jupiter entails many potential hazards in the course of the mission related to numerous small asteroids, many not detected, which will require on-board responses to avoid collisions. In addition, the long distance from the sun as the space vehicle moves outside the asteroid belt will require a nuclear power supply to power the vehicle for the 5 year mission. Because of the long duration, on board propellants will have to be used extremely efficiently and unplanned course corrections will need to be carefully planned to maintain sufficient fuel for attitude control over the full course of the mission. The possibility of robotic refueling for the attitude control systems going to and from Jupiter may be necessary to maintain reserves for the full duration of the mission. Communications with Earth will be limited due to distances from the Earth. At Jupiter the communication times will range from 35 minutes to 52 minutes based on the planets solar orbits. This long distance will require any decision which requires 75 minutes to 110 minutes to be made on board. Thus tactical re-planning will all be conducted on board. These tactical decisions can also affect options for strategic mission re-planning events such that decisions on board will have to consider strategic options in the tactical decisions.

VIII. Resupply Design Reference Mission

After clearing the orbital plane of the asteroid belt, the Marius continues on the first human exploration mission to Europa and Ganymede. As the mission proceeds into outer planetary space, a resupply ship, launched 3 months prior to the mission, is in a station keeping orbit. The Marius will rendezvous with this ship and take on additional food stores and fuels for the flight to Jupiter. Being 15 light minutes from Earth at the rendezvous point, the Marius must make all tactical decisions. As the Marius approaches the resupply ship, a caution and warning message is received from the ship indicating failure of steering thrusters on one side of the resupply vehicle. The Marius diagnostics function notifies the on-board mission management function that the resupply ship is unable to execute the planned docking maneuver. The crew activates a two body trajectory calculation to quickly re-calculate a rendezvous procedure accounting for the limited control available on the resupply ship as constraints on the new procedure. A new rendezvous timeline is produced based on these inputs and updated station keeping commands are sent to the resupply ship (Approach Activity) to accommodate the failed thruster bank. Modified rendezvous plans initiated by the crew via the HAL 9000 GNC Executive, are also determined for the Marius to ensure a safe approach to the resupply ship based on the new approach trajectory. A manual berthing operation is planned to assist in the safe capture and docking of the resupply ship as the planning system follows crew safety and vehicle integrity constraints for the rendezvous operation. The Space Exploration Vehicle (SEV) grappling arms will be used to guide the resupply ship to the docking hatch on the Marius. Once docked, the resupply ship is un-berthed using the SEV and placed a safe distance from the Marius as she continues toward Jupiter. The Marius will rendezvous with a second resupply ship, which has yet to be launched from Earth, on the return leg of the trip.


HAL 9000 Executive Components
HAL 9000 Execution Components
(1)
GNC Add New Activity
Crew
Timeline Generator
GNC New Plan Broadcast for Convergence
Convergence Processor
Plan Approval GNC Activity Added
Convergence Processor
New Plan to Memory for Execution Component
Synchronize Timeline Update
(2)
Propulsion Scans new Plan
Monitor Re-plan
Determine: Available Thrusters Thruster Time Vehicle Re-orientaton
Logic Modules
Resource Tracker
Core Linked List
Propulsion generates new plan
Timeline Generator
Propulsion New Plan Broadcast for Convergence
Convergence Processor
Propulsion Plan Approval
Convergence Processor
New Plan to Memory for Execution Component
Synchronize Timeline Update
(3)
Power Scans new Propulsion Plan
Monitor Re-plan
Determine: RPC Switches for Valve and Time for activation
Logic Modules
Resource Tracker
Core Linked List
Power generates new plan
Timeline Generator
Power: New Plan Broadcast for Convergence
Convergence Processor
Power Plan Approval
Convergence Processor
New Plan to Memory for Execution Component
Synchronize Timeline Update
(1)
GNC New Activity Timeline
(2)
Propulsion New Timeline
Timeline holds all thruster states and effective thruster times for all thrusters
Manual Execution
Auto Execution
(3)
Power New Timeline
Timeline holds all RPC States and effective on/off times for all RPC's
Manual Execution
Auto Execution
Crew: External Application Inputs

Note: Communications would also plan a potential transmitter switch due to the change in vehicle orientation, which then would lead to Power re-planning the RPC's for the switch in power amplifiers…. Communications could also plan the data recording and comm. configurations if the orientation did not provide a good line of sight for the transmitters.

Figure 7. Modified Rendezvous Planning Flow

IX. Jupiter Assured Crew Return

As the Marius approaches the Jovian system and prepares for the orbital insertion burn, there are many small objects detected by the collision detection and avoidance proximity radar. The structural integrity management function determines that these objects have sufficient energy to penetrate the Marius hull. In particular, a cloud of small asteroids has a direct collision course with the Marius near the orbital insertion point. At this point, quick tactical re-planning must take place. A new orbital trajectory is calculated for a new GN&C Activity with a slight change in planetary orbital inclination. The GN&C pointing and tracking function also provides new reference data for science observation objectives and modifications to the science instrument observation activities and plans are made. The orbit insertion at Europa is affected by the re-plan and new insertion targets are calculated to place the Marius in the proper orbit to maintain the options to choose between the primary and secondary landing sites. Once the orbital insertion maneuver for Jupiter has been safely completed, the crew begins observations and begins preparation for the Europa rendezvous. As the Marius orbits the planet many small objects are detected which require orbital maneuvers to avoid. The thruster firings for these maneuvers are longer and more frequent than the original mission plan had accommodated. The Monitor Logic Modules within the Propulsion Executive, adjusting for the longer and more frequent firings, notifies the crew that the Atitude Control System (ACS) fuel will be depleted before the planned mission departure time. The warning could have been annunciated during the re-plan/convergence of the maneuvers when the planned utilization exceeded the Mission Reserve as defined in the Mission Definition File. Due to the much more rapid rotation of the Earth than Jupiter around the Sun, the Jovian departure windows are 13 months apart corresponding to the Earths closest approach to Jupiter. The prognostics indicates the ACS will not last until the farer term departure window and the Marius must take the near term departure window which is two months after arrival. This requires a quick re-plan to accomplish the highest priority objectives, considering areas which minimize the projected use of the thrusters, and depart with sufficient reserves to complete the resupply rendezvous which will now be much closer to Earth (due to the early Jupiter departure) than originally planned. The Communications Executive notifies Mission Control of the re-plan, transmits the current vehicle state, and new rendezvous plans with the return leg resupply ship (which has not yet been launched) are transmitted back to the Marius accounting for an accelerated launch date for the resupply ship and the early return of the Marius. The new rendezvous point is now much closer to Earth than originally planned as the resupply ship no longer has time to arrive at the original coordinates before the Marius.

Figure 8. Resource Conflict Planning and Execution Flow

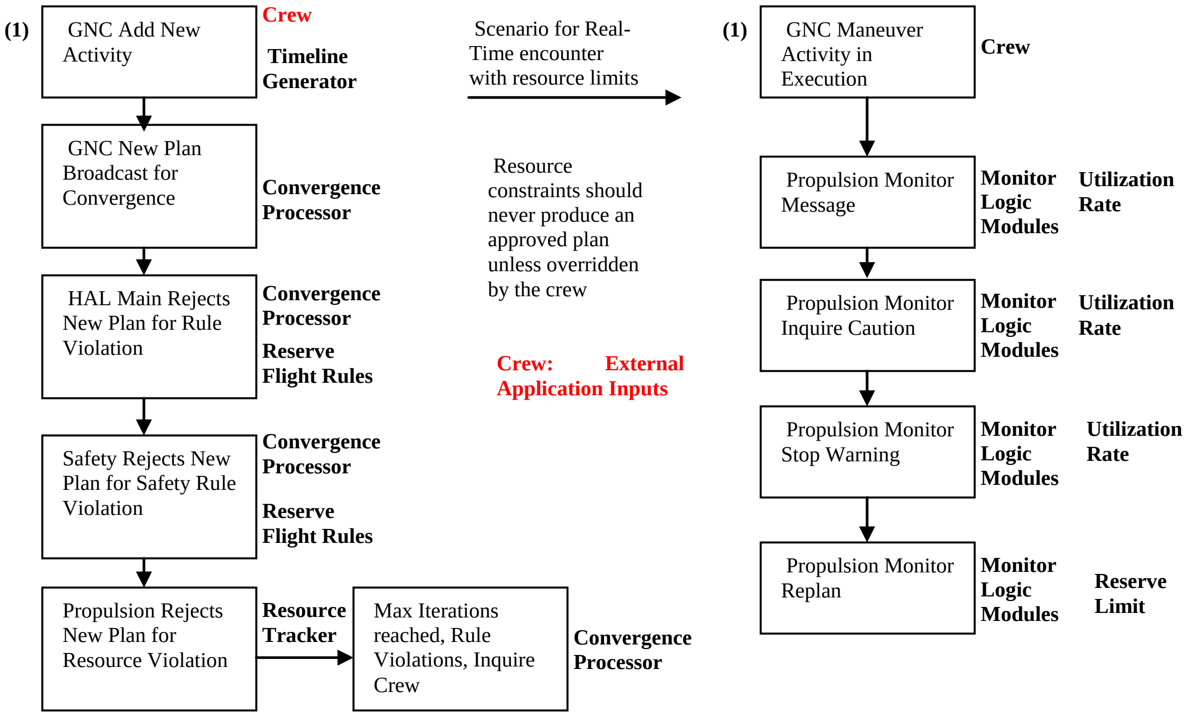
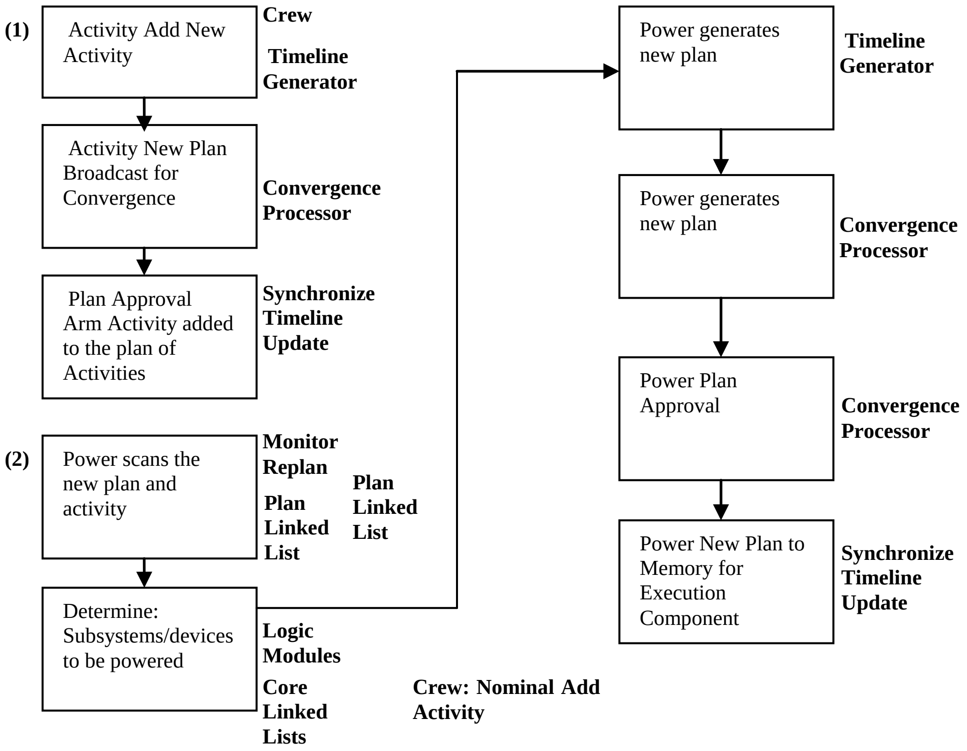

(1)
GNC Add New Activity
Crew
Timeline Generator
GNC New Plan Broadcast for Convergence
Convergence Processor
HAL Main Rejects New Plan for Rule Violation
Convergence Processor
Reserve Flight Rules
Safety Rejects New Plan for Safety Rule Violation
Convergence Processor
Reserve Flight Rules
Propulsion Rejects New Plan for Resource Violation
Resource Tracker
Max Iterations reached, Rule Violations, Inquire Crew
Convergence Processor
Scenario for Real-Time encounter with resource limits
Resource constraints should never produce an approved plan unless overridden by the crew
Crew: External Application Inputs
(1)
GNC Maneuver Activity in Execution
Crew
Propulsion Monitor Message
Monitor Logic Modules
Utilization Rate
Propulsion Monitor Inquire Caution
Monitor Logic Modules
Utilization Rate
Propulsion Monitor Stop Warning
Monitor Logic Modules
Utilization Rate
Propulsion Monitor Replan
Monitor Logic Modules
Reserve Limit

Figure 9. Maneuvering Arm Activity Planning Flow


(1)
Activity Add New Activity
Crew
Timeline Generator
Activity New Plan Broadcast for Convergence
Convergence Processor
Plan Approval Arm Activity added to the plan of Activities
Synchronize Timeline Update
(2)
Power scans the new plan and activity
Monitor Replan
Plan Linked List
Plan Linked List
Determine: Subsystems/devices to be powered
Logic Modules
Core Linked Lists
Crew: Nominal Add Activity
Power generates new plan
Timeline Generator
Power generates new plan
Convergence Processor
Power Plan Approval
Convergence Processor
Power New Plan to Memory for Execution Component
Synchronize Timeline Update

X. Conclusion

The HAL 9000 System is a very large design encompassing both the planning and execution of mission operations in an automated and autonomous manner. The main purpose of its design is to determine how to move mission control operations to on-board for a reduction or elimination of Earth based assets dependency when the time for manned deep space missions can be achieved. Being a large design, only segments of the system have been prototyped or developed for analysis and implementation. The Execution Component with Timeliner as the auto-procedure implementation has proven itself in the ISS manned space flight environment[1]. The Executive Planning Engine prototyping has shown that the architecture can perform the planning functions, although the system requires additional tools for analysis and input to crew decisions such as for course corrections, systems diagnosis and prognosis. Although the HAL 9000 System can operate the vehicle, it requires the crew for mission direction. It is not artificial intelligence but a design to incorporate specific intelligence into the planning and operation of a vehicle with minimal crew. The system design also provides the initial mechanism for crew autonomy and procedure development. The current Advanced Exploration Systems / Autonomous Mission Operations Project with the development work on a Habitat Demonstration Unit (HDU) / Deep Space Habitat (DSH) procedure execution simulation[3] advances our knowledge for requirements to pre-qualify crew authored procedures, by providing methods for testing each path of execution within an auto-procedure. Continued investigative development is required for the HAL Development Kit, Knowledge Pack interfaces and the numerous graphical user interfaces that are needed. Additional scenario testing is also required for further development and enhancement of the system overall and of the on-board simulations that would be needed to fully qualify a crew authored procedure before actual execution. Essentially, enough ground work has been accomplished at this point to raise the technical readiness of the system to the point where end to end prototyping of both software and hardware can be accomplished. The design reference missions that are described above show that additional analysis tools will be needed to assist the crew in real-time decisions and inputs to the HAL 9000 system. The design allows autonomous plan updates and implementation but further analysis on crew notification of plan changes is needed, such as plan updates that require mandatory crew notification (safety notifications, mission objective status). The identification of these tools and their requirements as well as the potential interfaces to the overall HAL architecture is needed.

Appendix A
Acronym List

ACS	Attitude Control System
DRM	Design Reference Mission
DSH	Deep Space Habitat
ECLSS	Environmental Control & Life Support System
EXE	Executable (software application)
FDDR	Fault Detection Diagnostics and Recovery
GN&C	Guidance Navigation and Control
HAL	Higher Active Logic
HDU	Habitat Demonstration Unit
ISS	International Space Station
MDF	Mission Definition File

References

[1] Stetson, H. K.; Deitsch, D. K.; Cruzen, C. A., Haddock, A. T.; "Autonomous Operations Onboard the International Space Station," IEEE Aerospace Conference, Big Sky, Montana, March 2007.

[2] Stetson, H. K.; Knickerbocker, G. K.; Cruzen, C. A., Haddock, A. T.; "The HAL 9000 Space Operating System," IEEE Aerospace Conference, Big Sky, Montana, March 2011.

[3] Haddock, A. T.; Stetson, H. K.; "Automated Operations Development for Advanced Exploration Systems," Space Operations 2012 Conference, Stockholm, Sweden, June 2012.

Biography

Howard K. Stetson *is a contractor for Marshall Space Flight Center, Space Systems Operations and is currently working as an analyst for the Space Launch Systems (SLS) flight operations, avionics and software, as well as the Advanced Exploration Systems-Autonomous Mission Operations project and has over 34 years of experience in software development and engineering. Preceding the SLS and AES projects, Mr. Stetson designed, developed and implemented the Higher Active Logic (HAL) autonomous system for ISS payloads. Mr. Stetson, an employee of Teledyne Brown Engineering, has received the NASA Space Flight Awareness Honoree Award, the Astronauts Personal Achievement Award (Silver Snoopy), and the NASA Exceptional Pubic Service Medal. Mr. Stetson is a member of the United States Selective Service System and the United States Parachute Association and currently has over 3000 jumps.*

Michael D. Watson *is the National Aeronautics and Space Administration (NASA) Space Launch System (SLS) Lead Discipline Engineer for Operations Engineering. He started his career with NASA developing International Space Station (ISS) Payload Training Complex and the ISS Payload Data Services System (PDSS). He also worked to develop remote operations support capabilities for the Spacelab Program installing Remote Operations Centers and services in the United States, Europe, and Japan. He subsequently served as Chief of the Optics Branch responsible for the fabrication of large x-ray telescope mirrors, diffractive optics, telescope systems. He served as Chief of the Integrated Systems Health Management (ISHM) and Sensors Branch and led a NASA team defining Vehicle Management System capabilities for human missions to Mars. His branch work included the definition of ISHM capabilities for the Ares family of launch vehicles. He graduated with a BSEE from the University of Kentucky in 1987 and obtained his MSE in Electrical and Computer Engineering (1996) and Ph.D. in Electrical and Computer Engineering (2005) from the University of Alabama in Huntsville.*

Ray Shaughnessy *is the Acting Deputy Manager for the Mission Operations Laboratory at The National Aeronautics and Space Administration, Marshall Space Flight Center located in Huntsville, Alabama. He has provided operations support for several NASA Programs including International Space Station, Orbital Space Plane and Constellation. He has a M.S. in Systems Engineering from the University of Alabama, Huntsville, a M.S. in Information Systems from Golden Gate University, San Francisco, Calif and a B.S. in Mining Engineering from the University of Kentucky.*