AMO EXPRESS: A Command and Control Experiment for Crew Autonomy Onboard the International Space Station

Howard K Stetson [1]
Howard.K.Stetson@nasa.gov
Teledyne Brown Engineering/NASA, Marshall Space Flight Center, Huntsville, AL 35812
Huntsville, Al 35812

Angie T. Haddock [2]
angie.haddock@nasa.gov
NASA, Marshall Space Flight Center, Huntsville, AL 35812

Jeremy Frank
Jeremy.d.Frank@nasa.gov
NASA Ames Research Center.
M/S 269-1
Moffett Field, CA 94035-1000
and
Randy Cornelius
James.r.cornelius@nasa.gov
Teledyne Brown Engineering/NASA, Marshall Space Flight Center, Huntsville, AL 35812
and
Lui Wang
Lui.wang-1@nasa.gov
NASA Johnson Space Center, Houston TX 77058
and
Larry Garner
Larry.w.garner@nasa.gov
Tietronix, Houston TX 77058

ABSTRACT

NASA is investigating a range of future human spaceflight missions, including both Mars-distance and Near Earth Object (NEO) targets. Of significant importance for these missions is the balance between crew autonomy and vehicle automation. As distance from Earth results in increasing communication delays, future crews need both the capability and authority to independently make decisions. However, small crews cannot take on all functions performed by ground today, and so vehicles must be more automated to reduce the crew workload for such missions.

NASA's Advanced Exploration Systems Program funded Autonomous Mission Operations (AMO) project conducted an autonomous command and control experiment on-board the International Space Station that demonstrated single action intelligent procedures for crew command and control. The target problem was to enable crew initialization of a facility class rack with power and thermal interfaces, and involving core and payload command and telemetry processing, without support from ground controllers. This autonomous operations capability is enabling in scenarios such as initialization of a medical facility to respond to a crew medical emergency, and representative of other spacecraft autonomy challenges. The experiment was conducted using the Expedite the Processing of Experiments for Space Station (EXPRESS) rack 7, which was located in the Port 2 location within the U.S Laboratory onboard the International Space Station (ISS). Activation and deactivation of this facility is time consuming and operationally intensive, requiring coordination of three flight control positions, 47 nominal steps, 57 commands, 276 telemetry checks, and coordination of multiple ISS systems (both core and payload). Utilization of Draper Laboratory's

[1] Computer Scientist, Space Systems Operations, NASA, Marshall Space Flight Center /MS 150, AIAA Non-Member.
[2] ASO Project Lead, Space Systems Operations, NASA, Marshall Space Flight Center /MS 150, AIAA Non-Member.
[3] Principle Investigator, Intelligent System Division, NASA Ames Research/MS 269-1, AIAA Non-Member.
[4] ISS Payload Rack Officer, Engineering Operations, NASA, Marshall Space Flight Center /MS 166, AIAA Non-Member.
[5] Insert Job Title, Department Name, Address/Mail Stop, and AIAA Member Grade for fourth author (etc).
[6] Insert Job Title, Department Name, Address/Mail Stop, and AIAA Member Grade for fourth author (etc).

Timeliner software, deployed on-board the ISS within the Command and Control (C&C) computers and the Payload computers, allowed development of the automated procedures specific to ISS without having to certify and employ novel software for procedure development and execution. The procedures contained the ground procedure logic and actions as possible to include fault detection and recovery capabilities. The autonomous operations concept includes a reduction of the amount of data a crew operator is required to verify during activation or de-activation, as well as integration of procedure execution status and relevant data in a single integrated display. During execution, the auto-procedures (via Timerliner) provide a step-by-step messaging paradigm and a high-level status upon termination. This messaging and high-level status is the only data generated for operator display. To enhance situational awareness of the operator, the Web-based Procedure Display (WebPD) provides a novel approach to the issues of procedure display and execution tracking. WebPD is a web based application that serves as the user interface for electronic procedure execution. It incorporates several aspects of the HTML5 standard. Procedures are written in a dialect of XML called Procedure Representation Language (PRL). WebPD tracks execution status in the procedure or procedures being displayed. WebPD aggregates and simplifies the auto-sequence execution status information, and formatted to be easily followed and understood by an operator who is not dedicated to actively monitoring the task. WebPD also provides an integrated data and control interface to pause or halt the execution in order to provide a check point of operation and to examine progress before starting the next sequence of activities. For this demonstration, the procedure was initiated and monitored from the ground. As the Timeliner sequences executed, their high-level execution status was written to PLMDM memory. This memory is read and downlinked via Ku-Band at a 1 Hz rate. The data containing the high-level execution status is de-commutated on the ground, and rebroadcast for WebPD consumption. A future demonstration will be performed onboard, with ISS astronauts initiating the operations instead of ground controllers. The AMO EXPRESS experiment demonstrated activation and de-activation of EXPRESS rack 7, providing the capability of future single button activations and deactivations of facility class racks. The experiment achieved numerous technical and operations 'firsts' for the ISS

I. INTRODUCTION

For over 50 years, NASA's crewed missions have been confined to the Earth-Moon system, where speed-of-light communications delays between crew and ground are practically nonexistent. The close proximity of the crew to the Earth has enabled NASA to operate human space missions primarily from the ground. This "ground-centered" mode of operations has had several advantages: by having a large team of the people involved on the ground, the on-board crew could be smaller, the vehicles could be simpler and lighter, and the mission performed for a lower cost.

Destination	Earth Distance (km)	1-way Time delay (s)
	38,400,000	1.3
NEOs (close)	variable	variable
Mars (close)	545,000,000	181.6
Mars (opposition)	4,013,000,000	1337.6

Table 1. Human spaceflight destinations in the Solar System, approximate distance from Earth, and approximate one-way light time delay.

NASA is now investigating a range of future human spaceflight missions that includes a variety of Martian destinations and a range of Near Earth Object (NEO) targets. These possibilities are summarized in Table 1. The table shows the approximate distance between the destination and the Earth, where the control center will be located, and the one-way light-time delay between the destination and Earth. As is evident from Table 1, future missions will be of much longer duration, and put crews much further from Earth, than today's missions. Accordingly, NASA has recently funded a number of projects to develop and test operations concepts for these future missions. Of significant importance is the balance between crew *autonomy* and vehicle *automation*. Future crews need both the authority to make decisions without inefficient communication back and forth with ground-based mission control. They will also need sufficient information and vehicle capability to make, and implement, those decisions. However, small crews cannot take on all functions performed by ground today, and so vehicles must be more automated to reduce the number of tasks crews are responsible for.

Current and prior work in this area includes crew autonomy experiments onboard ISS:

- *Crew Autonomous Procedures* [3]; astronauts performed numerous human spaceflight procedures without assistance from MCC, resulting in guidelines on the writing of these procedures.
- *The AMO experiment;* an ongoing experiment in transitioning responsibility for two ISS subsystems (water quality analysis hardware and crew laptops) to astronauts, with the aid of software tools.
- *Crew Self Scheduling;* a future experiment to evaluate tools for astronauts to schedule their own daily activity plans.

Previous work in this area also includes ISS automation:

- *Payload operations automation* [5]; A variety of ISS payloads were automated early in its operational lifetime.
- *Optimal Propellant Maneuvers and Zero Propellant Maneuvers* [2]; a special trajectory accomplished only using ISS' Control Moment Gyros (CMGs) was implemented using ground-based automation.

The experiment described in this paper demonstrates a transfer of authority from mission control to the crew in a similar spirit to the ISS Crew Autonomous Procedures experiment [3]. However, the system in question (an EXPRESS rack onboard ISS) has rich electronic command and telemetry interfaces, and thus is amenable to automation, as is used in previous payload autonomous operations.

Unlike prior work, the system in question spans both spacecraft core systems (power, thermal, and life support) and payloads (science). Also unlike prior work, the capability provided will ultimately allow crew to take as much control as desired (so-called adjustable autonomy).

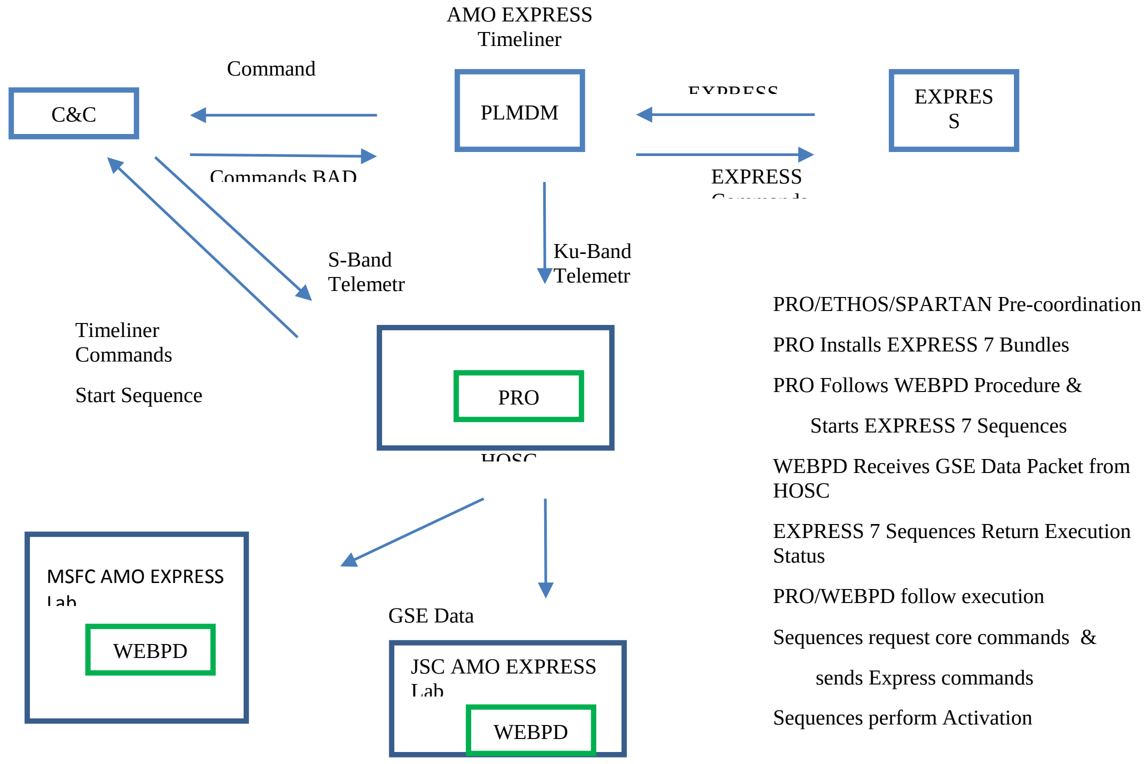
II. Operations Concept

The driving scenario for developing the operations concept is an unscheduled crew activation of a facility class rack, on a manned deep space mission with large communications delays, due to a crew medical emergency. With large communication delays, the crew would require the capability to autonomously activate the facility. Utilizing the ISS as a test-bed for such a command and control experiment, the EXPRESS Rack systems in the U.S. Lab were chosen as potential operating targets. The analysis of payloads resident within each EXPRESS Rack in the U.S. Lab, showed that EXPRESS Rack 7 was not continuously powered, and hence would provide a good target for autonomous activation and de-activation with no impact to payload operations. The analysis of ground command procedures that are required to be executed for EXPRESS rack operations and the division of responsibilities for activation/deactivation between MCC-H and Payload Operations Integration Center (POIC) resulted in an operations concept of intelligent functions that could be executed independently and with a single operator action. The division of the functions corresponds to the three mission operations positions responsible for executing the ground procedure, and also allows independent testing of the functions. The functions are Thermal Control, Power, Rack Boot-up Monitoring, Smoke Detection and Closeout functions for rack activation. For Rack De-activation, there would be only one operator action and hence one function to be executed. This concept would augment crew capabilities as it pertained to unplanned activation of a medical facility during a crew emergency. Automation onboard the ISS is accomplished by software running on MDMs (avionics computers). The concept required core commands to be sent from the Payload MDM (PLMDM) to the Command and Control MDM (CNCMDM) with the corresponding end item verification, as well as commands being sent to the PLMDM and EXPRESS Rack 7.

As the on-board scripting language for ISS, Timeliner was selected for the implementation of the concept. The Timeliner auto-procedures would be required to perform verification prior to sending a command as well as validation after sending the command. Fault detection would be embedded in the same manner as the Autonomous Fluid Transfer System [1], with recovery if possible. It must be noted that the EXPRESS systems were not designed for autonomous operations and it is not possible to recover from all failures.

The concept also includes a reduction of the amount of data a crew operator is required to verify during activation or de-activation, as well as integration of procedure execution status and relevant data in a single integrated display. During execution, the auto-procedures (via Timerliner) provide a step by step messaging paradigm and would provide a high-level status upon termination. This messaging and high-level status would be the only data generated for operator display. To enhance situational awareness of the operator, the Web-based Procedure Display (WebPD) provides a novel approach to the issues of procedure display and execution tracking. WebPD is under continuous development by NASA and has been employed in a variety of advanced technology experiments over the last several years[4,7]. Specifically, WebPD aggregates and simplifies the auto-sequence execution status information, and formatted to be easily followed and understood by an operator who is not dedicated to actively monitoring the task. WebPD also provides an integrated data and control interface to pause/ halt the execution in order to provide a check point of operation and to examine progress before starting the next sequence of activities. In this experiment, a two stage approach was chosen for WebPD execution, with the first a

“Follow Execution Only” approach, where the high-level status would simply be “followed” by WebPD on the ground, with the display of status and messages and the actual Timeliner sequences being started by the Payload Rack Officer via ground command. The second stage would involve WebPD being located on-board an ISS PCS attached to a payload local 1553 bus and commanding the Timeliner sequences, thus giving a changing locus of control to the crew for the same auto-procedures. The architecture diagram for the first stage operational concept is shown in Figure 1 below:



PRO/ETHOS/SPARTAN Pre-coordination

PRO Installs EXPRESS 7 Bundles

PRO Follows WEBPD Procedure &

Starts EXPRESS 7 Sequences

WEBPD Receives GSE Data Packet from HOSC

EXPRESS 7 Sequences Return Execution Status

PRO/WEBPD follow execution

Sequences request core commands &

sends Express commands

Sequences perform Activation

Figure 1 - AMO EXPRESS Context Diagram

III. Auto-Procedure Design

Prior experience with ISS Timeliner and the data and command interfaces provided the first hurdles that needed to be overcome before procedure development could begin. Not all core side commands and their end item verifications were available at the PLMDM, where Timeliner execution would take place. There were 5 thermal control commands, two remote power control module switch commands, two smoke detector commands and the corresponding end item telemetry for command verification that were required for execution from the core side console positions. All commands for the PLMDM and EXPRESS 7 were available to Timeliner negating any onboard changes on the payload side. With the identification of the missing commands and telemetry that were needed from the core side of ISS, we created and submitted two software change requests to the ISS Avionics Software Control Board (ASCB) to facilitate access to the missing commands and telemetry at the PLMDM as we began the procedure development. The ISS is essentially divided between core side operations and payload side operations which drives the command procedures to be "joint procedures" when multiple disciplines are involved. The initial design encompassed all the functions within one Timeliner bundle (executable file), which was later divided into two bundles, one for the core side functions (sequences) and one for the payload side functions (sequences). This turned out to be of great advantage as it separated the core side commands from the payload side commands and each could be tested separately at different facilities. This also enhanced procedure safety by further reducing the risk of inadvertent commanding. Each sequence performs one function as follows:

THERMAL Sequence – Rack Flow Control Assembly (RFCA) commanding and validation

POWER – Power Control Switch commanding and validation.

SMOKE – Smoke Detector commanding and validation.

INITDRIVES – Rack Boot-up Monitoring and rack validation

ACTCLOSEOUT – Rack activation closeouts (data downlinks)

DEACTIVATION – Complete rack deactivation commanding.

These are the main functional sequences. In addition, a TELEMETRYCHECK sequence was created to scan all the telemetry a Payload Rack Officer views to determine any anomalies after a rack boot-up. This sequence would return a count of errors detected, directing the Payload Rack Officer to view the EXPRESS 7 displays for the details. The MANUALCONFIG sequence was created to directly command the configuration of the rack for locker/drawer locations, communications and health and status when failures to both the Express Memory Unit (EMU) and Laptop occur and the Rack Interface Controller (RIC) configuration files could not be loaded. This command sequence becomes the one and only payload sequence that must be modified when the payload complement for the rack is changed.

During analysis of the ground command procedures, we noticed that there were a couple of failures within the procedures that could occur if the rack activation was not pre-coordinated with RFCA Software execution, Remote Power Control (RPC) Close/Open Inhibits, and the Rack Power Switch being in the off position. We had no control of whether the RFCA software was in execution, nor did we have or ask for the inhibit removal commands as we did not believe we could receive ISS Safety permission for these. In all of the cases, we could message the crew/ground for corrective action to the condition (which shouldn't exist if it was pre-coordinated), and monitor the end items for a period of time before terminating with an error or continuing on when corrected within the specified time. The real solution to removing these failure points is to obtain the Inhibit commands for inclusion in the sequences. In this way external interaction is further removed. The only hard failure we would not be able to work around programmatically would be the manual rack power switch, which requires a crew member to physically toggle, but in all cases when we encountered a detectable and correctable failure that the sequences could not correct themselves, the sequences message the action to take to the ground and crew, and monitor for the action to occur for 10 minute time periods. If the corrective action occurs, the sequence will continue, but if the corrective actions do not occur within the 10 minute monitoring period, the sequence would will, return a failure status code, and terminate.

There would be a specific order the sequences must be executed in and each sequence must complete and return with the single "GO" status return in order for the next sequence in the procedure to be started. Each sequence is re-entrant, so that if any errors occur which are not programmed for monitoring in the fault detection sections of the procedure, ground control could take-over, correct the condition and simply re-start the sequence that was in execution when the fault occurred. To identify all the known faults, analysis of all the payload anomaly reports was performed and also, it was decided that all devices/systems being commanded would be "state" checked prior to command transmittal and validated after command receipt. This added to the total number of faults being detected, but is required when performing autonomous commanding and autonomous procedures. One fall-out of this decision/requirement pertained to the increased number of status code returns from each sequence, but it did give detailed specific fault information to the operator.

Another hurdle we had to resolve was the high-level status codes the sequences would return from execution and how to transmit this data to the ground (and, eventually, to a computer onboard ISS running the WebPD). This was accomplished by receiving data written to a PLMDM memory allocation. This memory, called the HAL reserve, is write-accessible for Timeliner and downlinked automatically via Ku Band at a 1Hz rate. As this data is already being processed by the HOSC at Marshall Space Flight Center, no additional ground changes would be required to support the high-level status transmission from ISS. Since the data is written to PLMDM memory, the data would also be available for transmission to a 1553 connected Personal Computer System (PCS) onboard ISS, which would support the stage two operations concept. Transmittal of the data to the WebPD application is then a simple matter of defining a packet, listing the data to be transmitted and then executing the GSE Packet service provided as a standard HOSC user capability. A GSE packet is an Enhanced HOSC System / Consultative Committee for Space Data Systems (EHS/CCSDS) formatted data transmission. The data would be transmitted via Ethernet to specific Internet Protocol (IP) addresses and ports. After final development of the AMO EXPRESS Bundles, only 23 high-level status items were required, reducing the amount of telemetry an operator is required to look at for EXPRESS rack activation and deactivation by a very large amount.

IV. Development and Test

Procedure development began with translating the ground command procedures into Timeliner procedure sequences. The first function developed was the THERMAL control sequence and the identification of the Program Unique Identifier's (PUI's) to determine the Rack Flow Control Assembly (RFCA) software state (was the monitoring software started and in execution?), the current RFCA state (current mode?), sending the correct mode commands to initiate flow, determining when the flow rate was within range and finally monitoring the valve movement to command the lock-in of the flow rate. The flow rate set point utilized was 50 Kg per hour.

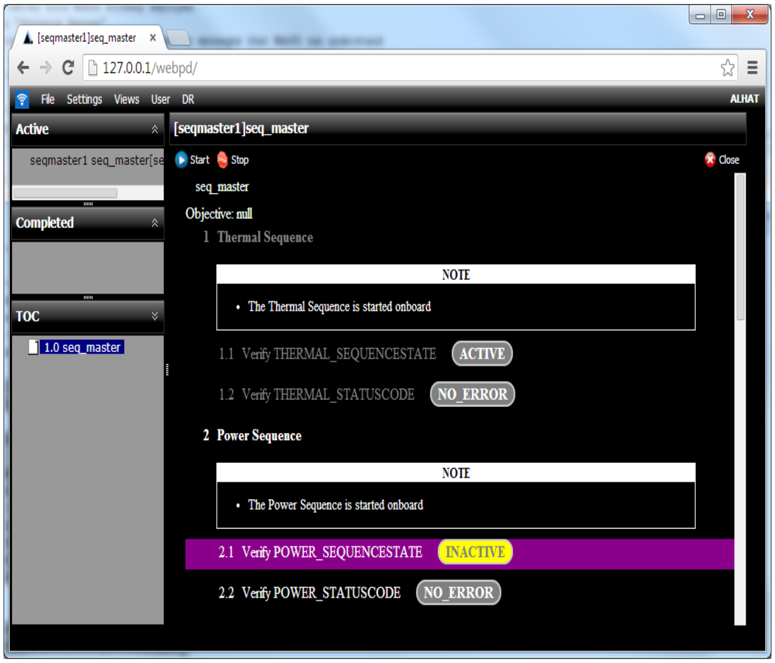
The second function to be developed was the POWER sequence which would apply power to EXPRESS Rack 7. This sequence verified the status of the specific remote power control module, and the Remote Power Control switch (RPC) before commanding the RPC to a closed position. Initially we were to power the rack via main and aux RPC's, but we encountered a bundle memory limit restriction (procedure too large) and the decision was made to only power via the Main RPC for procedure reduction.

The third function to be developed was the smoke detection enablement sequence, SMOKE. The sequence encompassed the verification of the smoke detection status, monitoring of the scatter and obscuration values and calculating the % trip value to determine if it was safe to command the smoke detector enablement. Enabling the smoke detector with obscuration and scatter limits exceeded would immediately sound a fire alarm.

The fourth function to be developed was the rack boot-up monitoring sequence, INITDRIVES. This was the first payload side sequence to be developed which monitored the boot-up from power-up that included monitoring the configuration file loads, retrieval drive operations and correcting any configuration anomalies. If no retrieval drives were available (such as a failed EXPRESS Memory Unit or failed or unattached Lap top), the Manual Configuration sequence would be initiated to command the configuration of the rack and its payloads.

After a successful rack boot-up, the Payload Rack Officer (PRO) performs many commanded actions for a final health and status check of the rack. These actions were developed into the Activation Closeout sequence, ACTCLOSEOUT, which would be the final sequence to be executed for rack activation. During auto-procedure development, the high-level status return codes were populated where each possible failure that was programmed for had a specific code, and only one success code would be defined for each sequence. In this way, the status return codes would point to a specific error condition and operations direction would be given for the response. Upon any status return, messages were generated in two downlink data streams providing informative descriptions of the error and the direct action to be taken by operators if any.

WebPD is a web based application that serves as the user interface for electronic procedure execution. It incorporates several aspects of the HTML5 standard. Procedures are written in a dialect of XML called Procedure Representation Language (PRL) [6]. WebPD operates in conjunction with PRLExecutive, which tracks execution status in the procedure or procedures being displayed. The WebPD procedure for AMO EXPRESS was developed at JSC and simply contained the sequences to be executed in "step" order. WebPD interfaces directly with PRLExecutive and indirectly with Timeliner via the GSE packet telemetry, as described in Figure 1. As a direct client to the PRLExecutive, WebPD is responsible for displaying procedure execution status as they are published. Also, when needed, PRLExecutive prompts the user for input. WebPD is responsible for displaying these user queries and providing a means of responding. Figure 4 below shows the WebPD following the execution of the 'seq master' procedure. Telemetry is displayed in the ovals, the dimmed lines are denoting completed steps, and the purple bar indicates the current location of the procedure execution.



[seqmaster1]seq_master

seq_master

Objective: null

1 Thermal Sequence

NOTE

- The Thermal Sequence is started onboard

1.1 Verify THERMAL_SEQUENCESTATE ACTIVE

1.2 Verify THERMAL_STATUSCODE NO_ERROR

2 Power Sequence

NOTE

- The Power Sequence is started onboard

2.1 Verify POWER_SEQUENCESTATE INACTIVE

2.2 Verify POWER_STATUSCODE NO_ERROR

FIGURE 2 – WEBPD DISPLAY

To serve AMO Express functions, WebPD monitors the executing Timeliner sequence by tracking the high-level status returns, and autonomously check off / follows the execution progress until the completion of the Timeliner sequence. The GSE data packet definition to support the WebPD application contained the program unique identifiers that the Timeliner sequences populated during execution providing the high-level status and messages. The packet definition is utilized by the HOSC to build and send data packets to remote installations. Our goal here was to define the data needed by WebPD to support both the ground "follow only" mode and the on-board "command" mode without making any changes to the data definition or the order of the data within the packet.

The WebPD application was delivered to MSFC to execute on a PC platform with Windows 7. The application reads CCSDS packet data over a UDP port. Once logged into the HOSC, an "AMO Operations" user initiates the packet generation and distribution of packets to the WebPD application via the standard HOSC GSE Operations capability. The IP addresses and data port identification becomes the only requirement once the packet content is defined. AMO operations defined and validated the packet definition and assigned a unique format identifier. The GSE Operations capability also allows the user the ability to "watch" packet transmission which provided and end to end monitoring capability of the data stream as each packet was displayed upon reception at the AMO workstation.

As the Timeliner sequences executed, their high-level execution status was written to PLMDM memory (HAL System memory). This memory is collected and downlinked via Ku-Band at a 1 Hz rate. The PUIs utilized by AMO and containing the high-level execution status is de-commutated at the HOSC and the values placed in the AMO data packet for WebPD consumption and distributed. WebPD requires the user to select the procedure to be executed, (in this case, the procedure simply follows the execution), and gives the user start/stop control. Once started, the AMO EXPRESS procedure reads the incoming data packets and displays the high-level execution status for each step in the procedure that is executed. No anomalies were ever encountered with the WebPD application, although a few enhancements will be submitted after AMO EXPRESS real-time operations is completed.

In addition to the real-time AMO EXPRESS Timeliner sequences, AMO operations created a WebPD TEST Sequence that would generate status data. This test sequence allowed testing of WebPD without having to allocate and reserve Core test assets as well as EXPRESS rack 7. Table 2 depicts the WebPD procedure step correlation to Timeliner Sequences where each step in the procedure has a corresponding Timeliner sequence to be executed.

Table 2. Procedure Step to Sequence Correlation

Step 1 Thermal	AMOE_THERMAL
Step 2 Power	AMOE_POWER
Step 3 Smoke Detection	AMO_SD_ENABLE
Step 4 Rack Boot Monitor	AMO_INIT_DRIVES
Step 5 Closeouts	AMOE_ACT_CLOSEOUT
Step 6 De-activation	AMOE_DEACTIVATION

The Single Button activation option for the PRO/Crew is the master sequence Activate_Express7 which starts each sequence in WebPD "step" order and verifies the return status before starting the next sequence in the procedure. WebPD would receive the high-level status from the sub-ordinate sequences the same as if they had been executed singularly. De-activation of EXPRESS 7 is a single step and is the final sequence to be executed for the experiment.

After a good procedure baseline was created of the logic flow and the core commands were received and would compile, we coordinated testing time. Test sessions were divided into the core side and the Express Rack (ER) (payload side). Core testing exercised use of the Rack Flow Control Assembly (RFCA), Rack Power Controller Module (RPCM), Rack Power Controller (RPC), and smoke detection enablement (SD).

Testing of the core bundle was exercised using the Software Development and Integration Laboratory (SIDL) at JSC. The SIDL can be configured by either using the Integrated Test Rig (ITR) or Hardware/Software Integration Lab (HSIL). Both facilities provide flight equivalent hardware replicating the onboard computers and network capabilities of the ISS. They contain a subset of the ISS architecture and can be configured to simulate connectivity of the Command and Control MDM (CNCMDM), Guidance Navigation and Control MDM, and the Payload Multiplexer-De-Multiplexer (PL MDM) as well as other combinations of Tier 2 and 3 MDM's. The SIDL does not provide access to an ER or ER equivalent. To test the ER or Payload software, we had to use facilities located at MSFC in the Payload Software Integration and Verification Facility (PSIVF) or the Payload Rack Checkout Unit (PRCU).

AMO never scheduled dedicated test time using the SIDL and always piggy backed on others testing objectives at JSC and MSFC. This was a decision to lessen the costs incurred by the project. By not having dedicated facilities for AMO to test and sharing resources and test objectives some days we only got to test for an hour or less during an 8 hour scheduled session. This was a challenging environment early on because most of the end item verifiers onboard were in raw counts and not calibrated data as observed on the ground.

The RFCA water flow counts, the set point temperatures, and the smoke detection scatter and obscuration telemetry values were in raw counts and not calibrated. The HOSC telemetry data base calibration coefficients were used to calculate the flow values and the set point values. Smoke detection was a little more of a challenge since all the smoked detection enablement had been handled by the JSC ETHOS console position. Working with ETHOS, we learned smoke detection was not enabled based on just scatter and obscuration, but a combination of both called "percent trip". ETHOS provided us with the calculation for percent trip and we used that derived value to determine whether we should enable smoke detection or not. With no way to manipulate the scatter or obscuration values in the models with SIDL, we wrote a C program that requested the scatter and obscuration values, calculated the percent trip and output it to the screen. Several real time rack scatter and obscuration values were obtained from ISS and PRO requested percent trip values from ETHOS. The values were input into the C program and we knew the percent trip values were correct when they matched what the ISS real time systems produced. After we knew we had good percent trip calculation, we could input different combinations of scatter and obscuration values to determine the upper acceptable threshold or enablement of smoke detection. Basically the same method of developing a C program to determine the raw count ranges to determine the RFCA set-point upper range and the flow meter ranges were used in combination with real time data dumps to verify.

Testing the ER bundle proved to be a little more challenging than the core side. The ER auto-procedures had to be tested in either PSIVF or the PRCU. After the bundles were compiled, we could not uplink them because critical ground servers for the command interface to MSFC did not exist. The file would have to be emailed to personnel responsible for the facilities configuration and then transferred to the PLMDM. After the files were copied to the PLMDM, PRO could downlink the PLMDM directory list file and update the Auto-Procedures Ground Management Tool (APGMT), which is the main command and control application for Timeliner operations.

Since the bundles were logically divided into core and payload/ER bundles, PRO could concentrate on the rack bundle. Not all lines of procedure steps could be tested because of system limitations. The ER subsystem flow for cooling and the thermal sensors located in the cooling lines had no real or simulated values associated with them.

We wrote C routines similar to the ones for the core side which accepted input ranges and output expected raw count values. The values were compared to Near Real Time (NRT) data from operating ISS racks. Nominally, all auto-procedures must have full execution path testing, but in the case of the thermal sensor testing, there was a small segment of the procedure that could not be tested completely, but the C program output matching real time NRT data, gave us a high confidence level the rack thermal sensor check will work well.

The PSIVF when using the qualification unit and the PRCU facility functioned identical as far as the Huntsville Operations Control Center (HOSC) interface with these facilities, even though, they were physically in 2 different locations. Both facilities consisted of a PLMDM functional equivalent unit (FEU) verified to flight data interfaces and running PEP flight software, an EXPRESS Laptop FEU running EXPRESS Rack Laptop flight software, a Payload Ethernet Hub Gateway (PEHG) FEU verified to flight interfaces (route 10Mbps Ethernet data from International Standard Payload Rack (ISPR) Payload simulations, verify traffic rates), and an Improved Payload Ethernet Hub Gateway (iPEHG) FEU verified to flight interfaces (route 100 Mbps Ethernet data from ISPR Payload simulations and external, non-simulated payloads). The PRCU and PSIVF provided a CNCMDM simulator interface. However, it is only used to pass commands to the PL MDM. Broadcast Ancillary Data and Payload Ancillary Data (BAD/PAD) are usually inhibited since there are not core systems available to pass data to the CNCMDM to broadcast. By inhibiting these CNCMDM broadcast, the PRO can command data loads for this core data into PL MDM memory, essentially providing a data simulation from the core side. This works well when setting up for an execution path test.

The HOSC provided the interface to SIDL, PSIVF, or the PRCU and received telemetry from the external facilities and issued commands to each of them. We never used a flight Mission Operational Mode Project (MOP) configuration to insure a full test environment. The MOP's provided for testing were always pre-mission MOP's which consisted of new software releases for either Enhanced HOSC System (EHS), PIMS, EHS PC (EPC), telemetry database or command database upgrades. As a result of the different combinations of HOSC upgrades, considerable time was lost while difficulties in system configuration issues were worked and eventually fixed, but all combinations of flight and ground software as well as databases were tested.

When the HOSC and our testing facility of the day were in harmony, test objectives consisted of running each bundle to completion and using Payload Event viewer and displays to verify expected end items or responses. The Auto-Procedures Ground Management Tool (APGMT) was used to issue Timeliner hold-at, sequence resume and sequence step commands in order to move in and out of Timeliner specific lines of the procedure. Not all end item verifications were provided in telemetry. When there were no telemetry verifiers, we used HAL internal memory Program Unique Identifier's to capture and downlink the desired value. When a procedure change was required, we had to retrieve the source file from PIMS to a local drive, edit and save the procedure back to the local drive, and then copy the procedure back into PIMS where it could be re-compiled. If we were in the PSIVF or using the PRCU, the byte swapped TLA and TLX file for each bundle we were testing would have to be emailed to the test engineer. If we were testing using the SIDL, we could drop the file in the MSFC drop box and uplink it after JSC retrieved the file to their command server. This latter uplink operation is flight like. The process for editing, compiling, and getting a bundle ready to be tested again, often took much longer than it took to modify the procedure. MSFC is currently working on moving the PIMS application to a share point server and will provide PIMS Timeliner edit capability. This will reduce procedure modification turnaround time significantly. The final testing encompassed the flight databases, MOP, EHS and flight software that would be encountered during real-time operations on-board the ISS.

Both test facilities, JSC and MSFC, provided challenging environments for AMO because there were always multiple objectives to be accomplished during the test and AMO was always the lowest priority. The biggest challenge was getting the facilities setup up in a correct configuration to run our tests. The SIDL had issues with properly configuring the PLMDM's and loading the Pre-Positioned Load's (PPL's) on the CNCMDM's for the Command Request Commands (CRC) required by AMO. The PRCU and PSIVF functioned well, although, the biggest challenge was keeping the data base current and keeping up with configuration changes such as, EHS and EPC software updates, and TIMELINER compiler changes. The personnel involved were professional and always worked hard to get AMO in a good state to be successful. Many times because of unforeseen issues with the test rig or MSFC hardware or software, hours and even days produced no productivity. If there was a test facility that combined the SIDL and PSIVF or PRCU into one integrated facility, testing time could be reduced to days or weeks verses many months.

V. Approval Process

Beginning in December 2012, the AMO-EXPRESS payload began the journey of development, payload manifestation, testing, and approvals from various ISS Boards/Management/Panels for on-orbit operations. The following will summarize the excursion through various ISS Boards/Management/Panels which AMO-EXPRESS had to receive approvals from in order to execute on-orbit.

In December 2012, the ISS as a Test-bed for Analog Research Joint Operations Panel (ISTAR JOP) supported AMO activities and received a "go-ahead" from the ISS Engineering Integration Working Group (EIWG). The following month, MSFC AMO presented the AMO-EXPRESS experiment idea to the MSFC Mission Operations Lab (MOL) Director and the Payload Operations Integration Facility (POIF) Managers and received a "go-ahead". During the period of January-February 2013, MSFC AMO met several times with Payload Rack Officers (PROs) to discuss MSFC AMO Overview & EXPRESS Rack Topology. After these meetings, it was determined EXPRESS Rack 7 would be the best EXPRESS Rack (ER) to use, for it is a non-continuous powered ER with only 1-2 Payloads. This decision would later impact the scheduling of AMO EXPRESS operations as during the development time, payloads would be manifested for this experiment rack.

During the month of March 2013, MSFC AMO presented the AMO-EXPRESS concepts and plans to EXPRESS Software Control Panel (ESCP). The ESCP did not have any concerns and gave AMO the "Go Ahead". MSFC AMO met with the EXPRESS Rack Operations Lead, PROs and Data Management Coordinators (DMCs) to summarize AMO-EXPRESS concepts and review technical impacts, and received no concerns/issues. During a telecon with the Payload Operations Directors (PODs), the AMO Project Manager, and the JSC Crew Office Chief Engineer, MSFC AMO briefed the Crew Office with the AMO-EXPRESS proposals. The JSC Crew Office Chief Engineer was supportive of the AMO-EXPRESS proposed plans, as long as they pass all boards/safety reviews, so AMO-EXPRESS received positive backing from the crew office.

In April 2013, the AMO Project Manager worked with the manager of the ISS Technology Demonstration Office to get AMO-EXPRESS listed in the Integrated Payload List (IPL). AMO-EXPRESS became manifested when it was listed in the 04/11/13 of the ISS Payloads Office IPL. The plans for AMO-EXPRESS were presented to the Technical Huntsville Operations Support Center (HOSC) Management Control Group (HMCG), and received approval for AMO-EXPRESS plans. AMO-EXPRESS Overview was presented to the ISS Research Planning Working Group (RPWG), and received no issues or concerns from this working group.

During June-August 2013, AMO-EXPRESS Change Evaluation Form (CEF) 4133 for Inc 39/40 was submitted and approved, the AMO EXPRESS Investigation Overview and Command Utilization was presented to the ISS NASA/Boeing Tagup, and a Payload Integration Agreement (PIA) for AMO-EXPRESS was submitted and approved. In August 2013, AMO-EXPRESS presented the AMO Ethernet Commanding Overview to the ISS Safety Review Panel (SRP), Special Topic Meeting. The result of this meeting was that the Payload Timeliner Process needed to be updated to include the process of issuing a Crit 2 Core Command in the Payload Timeliner Bundles, so AMO-EXPRESS worked with the ISS Computer Safety Panel and the Timeliner Operation Review Panel (TORP) and Safety Representative to get this change added. AMO-EXPRESS presented Software Change Request (SCR) 41920 (CCS PAD PPL to Support Payloads AMO EXPRESS) and SCR 41963 (PLMDM Command Requirements Required for AMO EXPRESS) to the ISS Software Change and Schedule Review Panel (SCRCP), and AMO-EXPRESS received approval to proceed to the Avionics Software Control Board (ASCB), where these SCRs were presented and approved.

From August 2013-May 2014, the development and testing of the AMO-EXPRESS Timeliner Bundles occurred. After development and successful testing of AMO-EXPRESS, the approvals from various ISS Board/Management/Panels for AMO-EXPRESS had to be done sequentially, so beginning in May 2014 and ending in October 2014, AMO-EXPRESS presented and received approval from the various ISS Boards/Management/Panels: Presented the AMO-EXPRESS Timeliner Bundles/Overview to the Timeliner Operations Review Panel (TORP), and received an approval for Timeliner Operations. Presented the AMO-EXPRESS Timeliner Bundles/Overview to the ISS Safety Review Panel (SRP), and received a "GO" for AMO-EXPRESS. The AMO-EXPRESS Timeliner Bundles/Overview was presented to the Payload Software Control Panel (PSCP), and received approval for AMO-EXPRESS and approval to proceed to the Avionics Software Control Board (ASCB) for review of AMO-EXPRESS. The AMO-EXPRESS Timeliner Bundles/Overview was presented to the MSFC Auto Procedure Control Panel (APCP), and received approval to proceed to submit an Engineering Change Request (ECR) to the Payload Operations Data File Control Board (PODFCB) for the uplink of the AMO-EXPRESS Operations Timeliner Bundles. Once the ECR was approved, the AMO-EXPRESS Overview was presented at the POIF Management Meeting, and received no issues or concerns. The AMO-EXPRESS Operations Overview was presented to the Avionics Software Control Board (ASCB) with no issues or concerns and received a “GO” to proceed. An Operations Change Request (OCR) was submitted after the ASCB meeting for the execution of the AMO-EXPRESS Operations. After the approval of the OCR, a Flight Note was issued for the AMO-EXPRESS Timeliner File Uplink, and the AMO-EXPRESS Summary was presented to the ISS Operations Tagup (Ops Tag) prior to execution of AMO-EXPRESS. This expedition through the ISS approval process from conception to on-orbit operations took 22 months.

VI. Real-Time Execution

Initial status of EXPRESS Rack 7: Powered, all payload locations un-powered, EXPRESS Laptop (ELC) was powered, flow rate set-point 77 Kg/Hr. PRO closed upper payload leg valve that supports MERLIN 5 cooling before operations.

GMT 308:15:00:47
PRO Installed two Bundles: AP_AMOE_PLSS and AP_AMOE_RACK

Run #1 Rack Deactivation, Single Action

GMT 308:15:32:40
PRO commanded the Start Sequence of AP_AMOE_PLSS.AMOE_DEACTIVATION

Verified all payloads un-powered
Verified all payload comm deactivated
Verified upper and lower payload leg valves are closed
Verified retrieval drive (EMU)
Commanded Lap Top Shutdown
Commanded Smoke Detector Monitoring inhibited
Commanded Rack Shutdown
Verified Health and Status data inactive
Verified the Main RPC State
Commanded Main RPC Open (remove power)
Verified RFCA State
Commanded RFCA to TEST State
Monitored Flow Rate

The sequence’s Flow Rate range was too tight and needs to be changed to a simple low value check. There were no impacts to the operations.

Verified Polling stopped and verified HAL6 was in execution. If HAL6 were not in execution, this sequence would have performed Shutdown Notification commanding and I/O inhibit commanding, but was not needed.

Run #2 Step wise EXPRESS Rack 7 Activation

GMT 308:15:52:23
PRO Commanded the Start Sequence of AP_AMOE_PLSS.AMOE_THERMAL

Verified RFCA Software in execution.
Verified RFCA State is in IDLE State
Verified RFCA to Flow Mode State
Commanded RFCA Flow Set-Point to 50 Kg/Hr
Commanded RFCA to Closed Loop Control
Monitored the Flow Rate
Flow Rate was obtained, and began monitoring the valve speed.
Verified Valve Speed of zero (not moving)
Commanded RFCA to IDLE State (Flow Rate Nominal)

GMT 308:15:56:42
PRO Commanded the Start Sequence of AP_AMOE_PLSS.AMOE_POWER

Verified PLMDM Polling and I/O enablement (HAL6 Actions)
Verified the Rack Power Switch is in the ON position
Verified the RPCM State (Operational)
Verified the Main and AUX RPC States (Open)

Verified Main RPC not Close Inhibited

Commanded Main RPC Close for Lab A Port 2 location

Verified PLMDM Health and Status Polling Started and Active

GMT 308:15:57:47

PRO Commanded the Start Sequence of AP_AMOE_PLSS.SDMON_ENABLE

Verified Obscuration Value

Verified Scatter Value

Calculated the % Trip Value within Enablement Range

Commanded Smoke Detector Enable

Verified Smoke Detector Enabled and BIT Check passed.

GMT 308:15:58:58

PRO Commanded the Start Sequence of AP_AMOE_RACK.AMOE_INIT_DRIVES

Monitored Boot Loader Active

Monitored for Health and Status received

Verified Retrieval Drive (EMU, which is currently in a failed state)

Commanded Laptop Power up

Laptop did not power on when commanded (command was potentially sent too soon after boot or MAC address was word swapped, we need to investigate)

TELEMTRY_CHECK Sequence executed autonomously

Verified the state of the rack and which configuration files were loaded or not loaded. TELEMETRY_CHECK sequence directed MANUAL_CONFIGURATION Sequence execution.

GMT 308:16:12:13

PRO Commanded the Start Sequence of AP_AMOE_RACK.AMOE_MANUAL_CONFIG

Verified the MCC heartbeat was incrementing and nominal.

Verified each of the 7 configuration files were loaded. For each configuration file that was not loaded, the manual configuration sequence commanded the RIC to the proper configuration. Because the rack did not have a retrieval drive available there were no configuration files that loaded.

Commanded each locker location configuration (8 lockers and 2 drawers)

Commanded the telemetry configuration for 2 payloads (GLACIER 4 and 5)

GMT 308:16:21:24

PRO Commanded the Start Sequence of AP_AMOE_RACK.AMOE_ACT_CLOSEOUT

Verified the state of the rack and commanded the Rack to operate mode

Commanded each of the 7 transmit statuses to the ground

Verified the thermal configuration to insure proper water flow through the rack and it failed the range check and halted execution. The rack ranges were acceptable and PRO resumed the sequence.

Verified all LAN and Sonic processing values were correct

Verified the AAA fan motor speed was in an acceptable range.

Verified all heartbeats (4) were incrementing and nominal.

Run #3 Rack Deactivation, Single Action

GMT 308:17:13:23

PRO commanded the Start Sequence of AP_AMOE_PLSS.AMOE_DEACTIVATION

Verified all payloads un-powered

Verified all payload comm deactivated

Verified upper and lower payload leg valves are closed

Verified retrieval drive (EMU)

Commanded Lap Top Shutdown

Commanded Smoke Detector Monitoring inhibited

Commanded Rack Shutdown

Verified Health and Status data inactive

Verified the Main RPC State

Commanded Main RPC Open (remove power)

Verified RFCA State

Commanded RFCA to TEST State

Monitored Flow Rate

The sequence's Flow Rate range was too tight and needs to be changed to a simple low value check. There were no impacts to the operations.

Verified Polling stopped and verified HAL6 was in execution. If HAL6 were not in execution, this sequence would have performed Shutdown Notification commanding and I/O inhibit commanding, but was not needed.

Run #4 EXPRESS Rack 7 Activation, Single Action

GMT 308:17:51:24

PRO Commanded the Start Sequence of AP_AMOE_PLSS.AMOE_ACTIVATE_EXPRESS7
(This sequence autonomously starts all other sequences based upon their status returns)
Commanded Start Sequence AP_AMOE_PLSS.AMOE_THERMAL
Verified RFCA Software in execution.
Verified RFCA State is in IDLE State
Commanded RFCA to Flow Mode State
Verified RFCA Flow Set Point at 50 Kg/Hr (set from previous Run #2)
Commanded RFCA to Closed Loop Control
Monitored the Flow Rate
Flow Rate was obtained, and began monitoring the valve speed.
Verified Valve Speed of zero (not moving)
Commanded RFCA to IDLE State (Flow Rate Nominal)
Commanded Start Sequence AP_AMOE_PLSS.AMOE_POWER
Verified PLMDM Polling and I/O enablement (HAL6 Actions)
Verified the Rack Power Switch is in the ON position
Verified the RPCM State (Operational)
Verified the RPC State (Open)
Verified Main RPC not Close Inhibited
Commanded Main RPC Close for Lab A Port 2 location
Verified PLMDM Health and Status Polling Started and Active
Commanded Start Sequence AP_AMOE_PLSS.SDMON_ENABLE
Verified Obscuration Value
Verified Scatter Value
Calculated the % Trip Value within Enablement Range
Commanded Smoke Detector Enable
Verified Smoke Detector Enabled and BIT check passed.
Commanded Start Sequence AP_AMOE_RACK.AMOE_INIT_DRIVES
Monitored Boot Loader Active
Monitored for Health and Status received
Verified Retrieval Drive (EMU, which is currently in a failed state)
Commanded Laptop Power up

Laptop did not power when commanded (command was potentially sent too soon after boot or MAC address was word swapped, we need to investigate)

TELEMTRY_CHECK Sequence executed autonomously

Verified the state of the rack and which configuration files were loaded or not loaded. TELEMETRY_CHECK sequence directed MANUAL_CONFIGURATION Sequence execution.

Commanded Start Sequence AP_AMOE_RACK.AMOE_MANUAL_CONFIG

Verified the MCC heartbeat was incrementing and nominal.

Verified each of the 7 configuration files were loaded. For each configuration file that was not loaded, the manual configuration sequence commanded the RIC to the proper configuration. Because the rack did not have a retrieval drive available there were no configuration files that loaded and the sequence issued all the commands necessary to put the rack in a nominal configuration.

Commanded Start Sequence AP_AMOE_RACK.AMOE_ACT_CLOSEOUT

Verified the state of the rack and commanded the Rack to operate mode

Commanded each of the 7 transmit statuses to the ground

Verified the thermal configuration to insure proper water flow through the rack and it passed the

range check on this occurrence. The rack ranges were acceptable.
Verified all LAN and Sonic processing values were correct
Verified the AAA fan motor speed was in an acceptable range.
Verified all heartbeats (4) were incrementing and nominal.

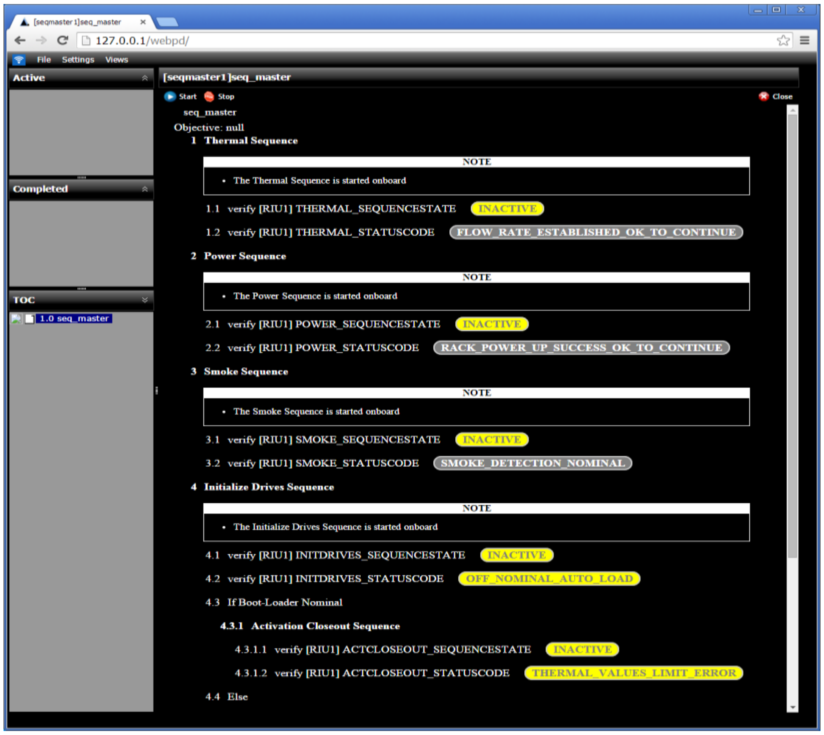
GMT 308:18:23:18

PRO Commanded Timeliner Bundle Removal's (2 bundles)

GMT 308:18:23:51

PRO Commanded Timeliner Bundle Deletions (4 Files, executables and address map files) and verified.

As shown in Figure 1, real-time execution of the sequences onboard was initiated by an operator on the ground with the WebPD application following the execution status. The progress of the sequences were monitored from both the HOSC at Marshall Spaceflight Center, and also from Johnson Space Center. The WebPD application is dependent upon a packet of predefined data downlinked from ISS via Ku-Band. The data is forwarded to WebPD from the HOSC utilizing the GSE Operations capability within the Enhanced HOSC System (EHS). AMO defined the data to be downlink and the Timeliner Sequences populated messages, status return codes and active/inactive status within this downlink. The Timeout feature of WebPD made it appear that the procedure was in error before execution actually begun. The procedure is also expecting to be executed in step order, which did not occur due to the powered state of EXPRESS 7. There were large time breaks between status return codes, which left the user "wondering" what was currently in operation. After the first de-activation and sub-sequent activation. WebPD was initialized for a clean run prior to the second de-activation and first single action activation. AMO left WebPD in execution and did not initiate a WebPD procedure start. This allowed WebPD to simply follow and display the status returns codes. Below in figure 3 and 4 are the WebPD results of Run #3. Since WebPD was not started, ERROR frames were not generated on the display. Overall, WebPD followed each step whether in step order or not and correctly interpreted the status return codes from the Timeliner sequences. The next step in AMO EXPRESS operations will have the real-time execution of the sequences being initiated from WebPD on-board by the crew.



[seqmaster1]seq_master

seq_master

Objective: null

1 **Thermal Sequence**

NOTE
- The Thermal Sequence is started onboard

1.1 verify [RIU1] THERMAL_SEQUENCESTATE INACTIVE

1.2 verify [RIU1] THERMAL_STATUSCODE FLOW_RATE_ESTABLISHED_OK_TO_CONTINUE

2 **Power Sequence**

NOTE
- The Power Sequence is started onboard

2.1 verify [RIU1] POWER_SEQUENCESTATE INACTIVE

2.2 verify [RIU1] POWER_STATUSCODE RACK_POWER_UP_SUCCESS_OK_TO_CONTINUE

3 **Smoke Sequence**

NOTE
- The Smoke Sequence is started onboard

3.1 verify [RIU1] SMOKE_SEQUENCESTATE INACTIVE

3.2 verify [RIU1] SMOKE_STATUSCODE SMOKE_DETECTION_NOMINAL

4 **Initialize Drives Sequence**

NOTE
- The Initialize Drives Sequence is started onboard

4.1 verify [RIU1] INITDRIVES_SEQUENCESTATE INACTIVE

4.2 verify [RIU1] INITDRIVES_STATUSCODE OFF_NOMINAL_AUTO_LOAD

4.3 If Boot-Loader Nominal

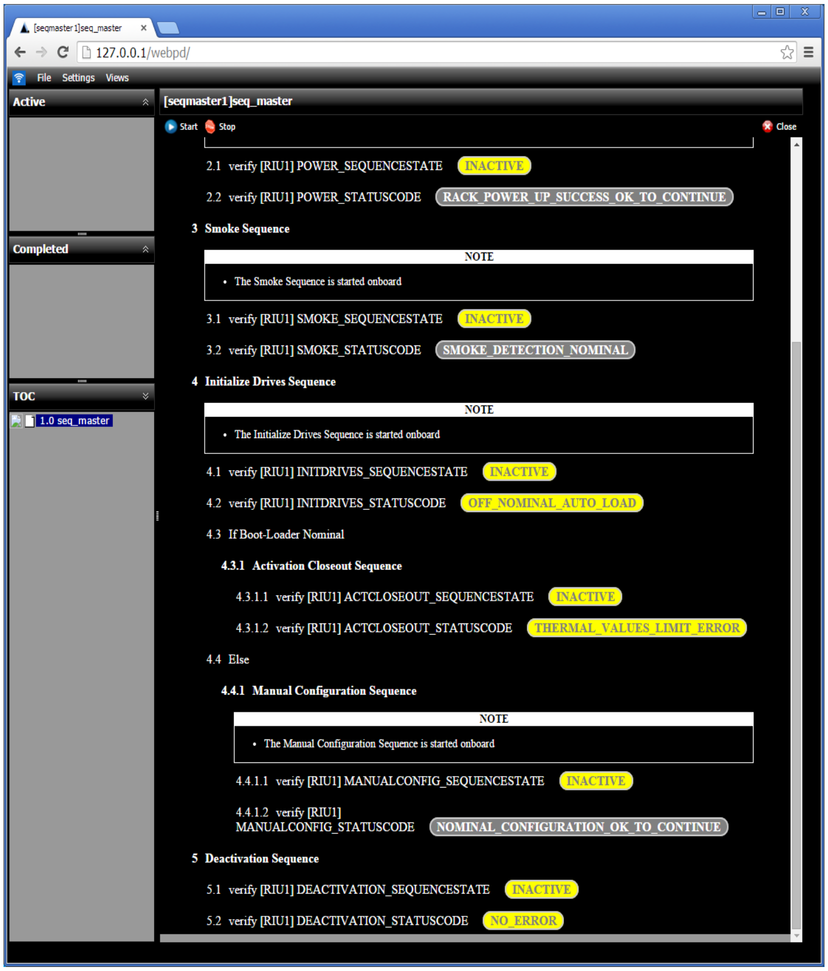
4.3.1 Activation Closeout Sequence

4.3.1.1 verify [RIU1] ACTCLOSEOUT_SEQUENCESTATE INACTIVE

4.3.1.2 verify [RIU1] ACTCLOSEOUT_STATUSCODE THERMAL_VALUES_LIMIT_ERROR

4.4 Else

Figure 3 WeBPD Real-Time Results



[seqmaster1]seq_master

2.1 verify [RIU1] POWER_SEQUENCESTATE INACTIVE

2.2 verify [RIU1] POWER_STATUSCODE RACK_POWER_UP_SUCCESS_OK_TO_CONTINUE

3 Smoke Sequence

NOTE

- The Smoke Sequence is started onboard

3.1 verify [RIU1] SMOKE_SEQUENCESTATE INACTIVE

3.2 verify [RIU1] SMOKE_STATUSCODE SMOKE_DETECTION_NOMINAL

4 Initialize Drives Sequence

NOTE

- The Initialize Drives Sequence is started onboard

4.1 verify [RIU1] INITDRIVES_SEQUENCESTATE INACTIVE

4.2 verify [RIU1] INITDRIVES_STATUSCODE OFF_NOMINAL_AUTO_LOAD

4.3 If Boot-Loader Nominal

4.3.1 Activation Closeout Sequence

4.3.1.1 verify [RIU1] ACTCLOSEOUT_SEQUENCESTATE INACTIVE

4.3.1.2 verify [RIU1] ACTCLOSEOUT_STATUSCODE THERMAL_VALUES_LIMIT_ERROR

4.4 Else

4.4.1 Manual Configuration Sequence

NOTE

- The Manual Configuration Sequence is started onboard

4.4.1.1 verify [RIU1] MANUALCONFIG_SEQUENCESTATE INACTIVE

4.4.1.2 verify [RIU1] MANUALCONFIG_STATUSCODE NOMINAL_CONFIGURATION_OK_TO_CONTINUE

5 Deactivation Sequence

5.1 verify [RIU1] DEACTIVATION_SEQUENCESTATE INACTIVE

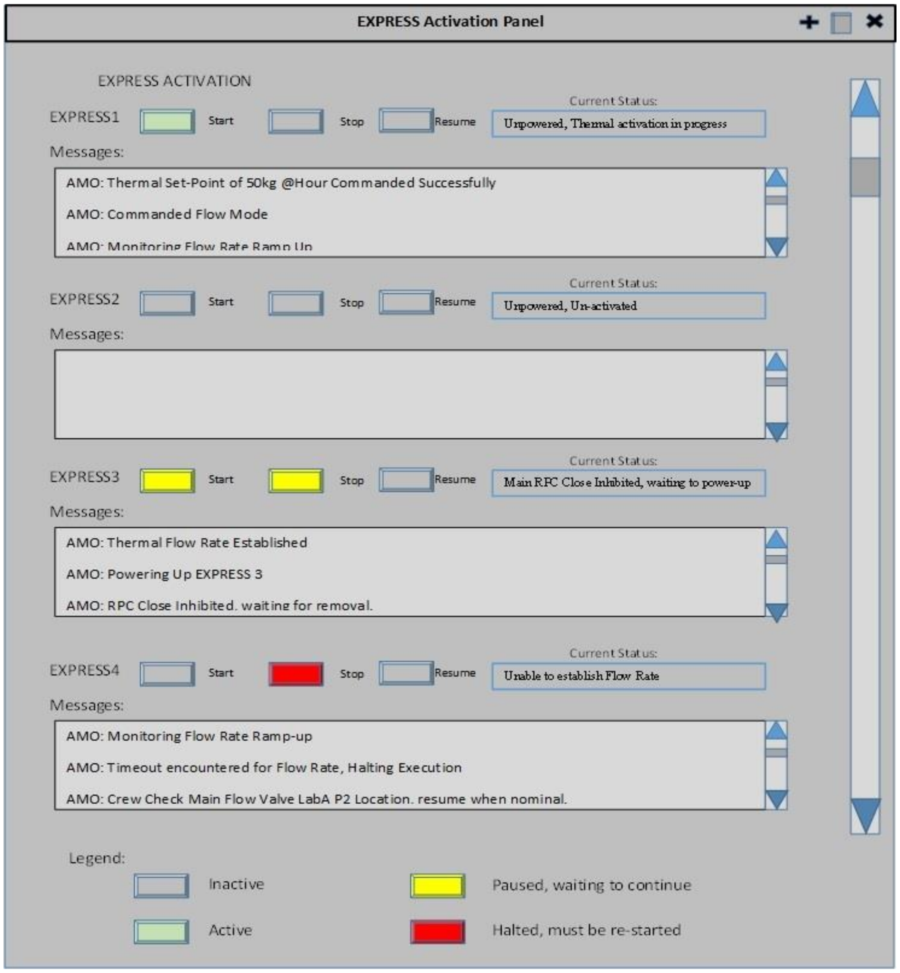
5.2 verify [RIU1] DEACTIVATION_STATUSCODE NO_ERROR

Figure 4 WeBPD Real-Time Results

VII. Summary and Future Work

The successful development and implementation of intelligent auto-procedures has proven that a significant spacecraft control function can be transferred from ground to onboard. More important, the demonstration shows that this function can be automated without incurring a significant increase of crew tended manual procedures. The single action activation and deactivation of the facility rack EXPRESS 7 shows that the current paradigm of console operator command procedures being duplicated by on-board auto-procedures can render the manual procedure to be obsolete. This is especially true when the single action combines the procedure operations of multiple console positions. If a command procedure and its volume of steps can be reduced to a single step, the result is a single step procedure, which essentially becomes a single button function and no longer in the “procedure” realm at the operations level.

The next question becomes how much information is required to be delivered to the crew during the autonomous execution of the function and the content of this execution status. The AMO EXPRESS experiment will eventually lead to the crew having the capability to initialize and power a facility class rack, which is a capability they do not have today. Figure 5 depicts a notional EXPRESS rack operations panel that provides the crew with single action EXPRESS activation. The panel is scrolled to uncover the complete suite of EXPRESS racks and each EXPRESS segment provides for the current status of the rack as well as all the messaging from the auto-procedures that perform the activation functions. An equivalent De-activation panel would be required to encompass both activation and de-activation.



EXPRESS1 — Current Status: Unpowered, Thermal activation in progress

Messages:

AMO: Thermal Set-Point of 50kg @Hour Commanded Successfully

AMO: Commanded Flow Mode

AMO: Monitoring Flow Rate Ramp Up

EXPRESS2 — Current Status: Unpowered, Un-activated

Messages:

EXPRESS3 — Current Status: Main RPC Close Inhibited, waiting to power-up

Messages:

AMO: Thermal Flow Rate Established

AMO: Powering Up EXPRESS 3

AMO: RPC Close Inhibited. waiting for removal.

EXPRESS4 — Current Status: Unable to establish Flow Rate

Messages:

AMO: Monitoring Flow Rate Ramp-up

AMO: Timeout encountered for Flow Rate, Halting Execution

AMO: Crew Check Main Flow Valve LabA P2 Location. resume when nominal.

Legend:

□ Inactive

🟩 Active

🟨 Paused, waiting to continue

🟥 Halted, must be re-started

Figure 5. Notional EXPRESS Activation Panel

During AMO EXPRESS operations, the auto-procedures produced ASCII messages for operator viewing and the messages essentially detailed what the auto-procedures were performing and encountering. Is this enough information to the crew? Is a drill down capability to lower level displays or procedure sub-steps needed? Or is a drill down to a detailed maintenance procedure needed when anomalies occur that the auto-procedures were not programmed to correct or recover from? During development of AMO EXPRESS, we encountered varying ways to display and control EXPRESS rack operations, from the in use today PRO displays and commands, to the WebPD procedure display, the Auto-Procedures Ground Management Tool HOSC application that controls Timeliner operations directly, test displays used for direct telemetry value viewing, network packet displays used for viewing raw data downlinked via Ku-Band, and this notional panel display. Each provides a different view of the same operations but which is best? Or are there parts of one display that combined with another provides the best information to the crew? This is part of a current study for determination. The results will most certainly be interesting.

Appendix A
Acronym List

AES	-	Advanced Exploration Systems
AMO	-	Autonomous Mission Operations
APCP	-	Auto Procedure Control Panel
APGMT	-	Auto Procedure Ground Management Tool
ASCB	-	Avionics Software Control Board
ASCII	-	American Standard Code for Information Interchange
BAD	-	Broadcast Ancillary Data
CEF	-	Change Evaluation Form
CCSDS	-	Consultative Committee for Space Data Systems
CMG	-	Control Moment Gyro
CNCMDM	-	Command and Control Multiplexer De-multiplexer
CRC	-	Command Request Commands
DMC	-	Data Management Coordinators
ECLSS	-	Environmental Control and Life Support Systems
EIWG	-	(ISS) Engineering Integration Working Group
EHS	-	Enhanced HOSC System
EMU	-	EXPRESS Memory Unit
EPC	-	EHS PC
ESCP	-	EXPRESS Software Control Panel
ETHOS	-	JSC Console Position
EXPRESS	-	Expedite the Processing of Experiments for Space Station
FEU	-	Functionally Equivalent Unit
GCP	-	Ground Control Procedure
GSE	-	Ground Support Equipment
H&S	-	Health and Status
HAL	-	Higher Active Logic
HSIL	-	Hardware/Software Integration Lab
HMCG	-	HOSC Management Control Group
HOSC	-	Huntsville Operations Support Center
HTML	-	Hypertext Markup Language
IP	-	Internet Protocol
IPL	-	Integrated Payload List
ISPR	-	International Standard Payload Rack
ISS	-	International Space Station
ISTAR	-	ISS as a Testbed for Analog Research
ITR	-	Integrated Test Rig
JSC	-	Johnson Space Center
JOP	-	Joint Operations Panel
MCC	-	Mission Control Center
MDM	-	Multiplexer De-multiplexer
MSFC	-	Marshall Space Flight Center
NEO	-	Near Earth Object
NRT	-	Near Real Time
OCR	-	Operations Change Request
PAD	-	Payload Ancillary Data
PEHG	-	Payload Ethernet Hub Gateway
PEP	-	Payload Executive Processor
PCS	-	Personal Computer System

PIMS	-	Payload Information Management System
PLMDM	-	Payload Multiplexer De-multiplexer
POD	-	Payload Operations Director
PODFCB	-	Payload Operations Data File Control Board
POIC	-	Payload Operations Integration Center
POIF	-	Payload Operations Integration Function
PPL	-	Pre Positioned Loads
PRL	-	Procedure Representation Language
PRO	-	Payload Rack Officer
PRCU	-	Payload Rack Checkout Unit
PSIVF	-	Payload Software Integration and Verification Facility
PUI	-	Program Unique Identifier
RFCA	-	Rack Flow Control Assembly
RIC	-	Rack Interface Computer
RPC	-	Rack Power Controller
RPCM	-	Rack Power Controller Module
RPWG	-	Research Program Working Group
SCR	-	Software Change Request
SCRCP	-	(ISS) Software Change and Schedule Review Panel
SDIL	-	Software Development and Integration Laboratory
SRP	-	(ISS) Safety Review Panel
SSC	-	Station Support Computer
TORP	-	Timeliner Operations Review Panel
UDP	-	User Datagram Protocol
WebPD	-	Web –Based Procedure Display
XML	-	Extensible Markup Language

Appendix B

References

Periodicals

1Haddock, A. T., Stetson, H. K. "Advancing Autonomous Operations for Deep Space Vehicles". Proceedings of the 13th AIAA International Conference on Space Operations Conference, Los Angeles CA., 2014

2Bedrossian N., Bhatt, S.A., Lammers M., Nguyen, L., Zhang, Y. "First Ever Flight Demonstration of Zero Propellant Maneuver Attitude Control Concept". Proceedings of the AIAA Guidance, Navigation and Control Conference, Hilton Head, SC; 2007

3Beisert, S., Rodriggs, M., Moreno, F., Korth, D., Gibson, S., Lee, Y. H., Eagles, D. E. "Development and Execution of Autonomous Procedures Onboard the International Space Station to Support the Next Phase of Human Space Exploration". Proceedings of the AIAA Space Conference, San Diego, 2013.

4Kortenkamp, D., R.P., Schreckenghost, D., Wang, L., "Developing and Executing Goal-Based, Adjustably Autonomous Procedures" AIAA InfoTech@Aerospace Conference, AIAA, Ronhert Park, CA 2007

5Stetson, H.K., Deitsch, D.K., Cruzen, C.A., Haddock, A.T. Autonomous Payload Operations Onboard the International Space Station. Proceedings of the IEEE Aerospace Conference, 2007.

6Kortenkamp, D., Bonasso, R. P., Schreckenghost, D., Dalal, K. M., Verma, V., Wang, L.. "A Procedure Representation Language for Human Spaceflight Operations". Proceedings of the International Conference on Environmental Systems, 2008.

7Frank, J., Spirkovska, L., McCann, R., Wang, L., Pohlkamp,K.,Morin, L. "Autonomous Mission Operations". Proceedings of the IEEE Aerospace Conference, 2013.