International Space Station (ISS) Payload Autonomous Operations Past, Present and Future

Randy Cornelius[1]
Teledyne Brown Engineering, Huntsville, AL 35801

and

Jeremy Frank[2]
NASA Ames Research Center, Moffett Field, CA 94035-1000

Abstract

Draper Laboratory's Timeliner is a scripting and automation system that runs onboard computers in the International Space Station (ISS). Timeliner is fully integrated with ISS and can be used to automate ISS operations tasks. Some of the most challenging aspects of operating a payload in low earth orbit are communication delays, ground equipment failures, and human errors. How does a Payload Developer (PD) know their equipment is operating nominally and collecting science in the most efficient way possible or even powered at any given time? During a ground Loss of Signal (LOS) data outage, PD's have no insight into their experiments state, and benefit greatly from Timeliner scripts executing on ISS to perform telemetry monitoring and commanding operations. This paper will discuss current software designs, and new operational uses for Timeliner. Existing Timeliner capabilities discussed will include: autonomous EXPRESS Rack activation and deactivation; autonomous science data downlinks; Minus Eighty Degree Freezer (MELFI) Dewar autonomous safing; JAXA and ESA module autonomous payload facility safing; as well as many others. New operational concepts discussed will include allowing Timeliner on the payload computer to issue core commands (Thermal, Power, Fire Detection), creation of new ground tools that will monitor the current status of Payload Racks as well as all the messaging from autonomous scripts executing, decreasing approval time for Timeliner bundles, and opening up the Payload MDM Enhanced Processor & Integrated Communications Card (EPIC) interface. The EPIC interface could provide a new crew interface for PL Timeliner execution. The new interface could operate on either a Payload Computer System (PCS) or a Station Support Computer (SSC) that is plugged into either the Payload LAN or the Operations LAN which will make communicating to the PL MDM more flexible and greatly increase band width for communication.

I. Introduction

In April 2016, the International Space Station (ISS) laboratory module Destiny will have supported science research in Low Earth Orbit (LEO) for over fifteen years. Since its activation in February 2001, flight controllers at the NASA-Marshall Space Flight Center (MSFC) in Huntsville Alabama's ISS Payload Operations and Integration Center (POIC) operate scientific experiments and provide payload support to Payload Developers (PD). Their systems are complex and require a great deal of expertise and engineering know-how to operate effectively and safely [2]. See Figure 1.

---

[1] Payload Rack Officer and Timeliner Operations Lead,, NASA/MSFC

[2] NASA Autonomous Systems Operations (ASO) Project Manager, NASA/ARC

Figure 1. **Payload Operations and Integration Center (POIC) control room.** *The POIC is the main control room for ISS payload operations. The Payload Operations Director (POD) is in charge of the room, Operations Controller (OC) who supports daily crew activities, Data Management Coordinator (DMC) who is responsible for ISS video and data systems, and the Payload Rack Officer (PRO) who configures and supports NASA owned facilities and payloads make up the main body of controllers.. All those positions work 24/7.*

A supporting technology employed to ensure the scientific and engineering data generated by the systems onboard are handled properly is a software tool commonly referred to as "Timeliner" and it is currently the only man rated scripting language available today. During the early years of ISS operations, ground controllers were routinely sending dozens of commands daily just to configure the facilities, data systems, activate and deactivate experiments, and many other routine, complex, yet necessary activities in order to successfully utilize the ISS as a world-class science facility. When operations were first supported there were few payloads and experiments that had to be managed [3]. As ISS grew to assembly complete, the payloads and facilities grew with it along with science responsibilities. As such, Timeliner development grew to help payloads operate more efficiently and ensure operational requirements were being met when ground operators did not have communication with ISS.

Fortunately, NASA had anticipated the increase in use and had planned to use the Timeliner User Interface Language (UIL) on ISS to alleviate communication related issues with payloads and facilities. The Timeliner UIL was developed by the Charles Stark Draper Laboratory in 1981 [1] for use in simulating tasks performed by astronauts aboard the Space Shuttle. In 1992, Timeliner was selected by NASA as the UIL for the ISS, and it was incorporated into the Command and Control Multiplexer-DeMultiplexer (C&C MDM) and the Payload MDM (PL MDM). See Figure 2. Timeliner is used on the C&C MDM to command to core ISS services (power, thermal, life support, etc.) and on the PL MDM to command to the PL MDM, facilities and payloads on ISS.

Some of the most challenging aspects of operating a payload in low earth orbit are communication delays, ground equipment failures, and human errors. How does a Payload Developer (PD) know their equipment is operating nominally and collecting science in the most efficient way possible or even powered at any given time? During a ground Loss of Signal (LOS) data outage, PD's have no insight into their experiments state. Autonomous payload operations utilizing Timeliner on-board the ISS has made operating the Payload segment more efficient for ground operators, saving both time and science. PDs benefit greatly from Timeliner scripts executing on ISS to perform telemetry monitoring and commanding operations during LOS conditions.

This paper will discuss current software designs, and operational uses for Timeliner scripts that operate on the ISS Payload MDM (PL MDM) to make facility and payload operations easier and more reliable. Timeliner capabilities discussed will include AMO EXPRESS that autonomously activate and deactivate EXPRESS rack 7, MAMS daily downlink that autonomously downlinks and resets a 24 hour buffer of data at GMT 18:00 every day, Minus Eighty Degree Freezer (MELFI) Dewar monitor sequences that will terminate operation of a dewar that experiences an out of state temperature, JEM Safe and COL Safe bundles that will power off all NASA facilities in the JEM and JAXA modules during a vehicle emergency, Payload MDM Real Time monitor bundle that monitors all NASA owned Remote Power Controllers (RPC) to start payload health and status processing and services, install payload Timeliner bundles when needed, and the Higher Active Logic (HAL) system that initializes the Timeliner system and will eventually manage and monitor ISS payload rack operations.

NASA is engaged in continued development of automation, both to improve ISS and also explore concepts needed to eventually carry humans to Mars. New operational concepts discussed will include opening up the pathway between the PL MDM and Tier 1 computers for core commanding (Thermal, Power, Fire Detection) that is currently only accomplished by ground operators, creation of ground tools that will monitor the current status of Payload Racks as well as all the messaging from autonomous scripts executing or ready to perform actions, decreasing approval time for Timeliner bundles that contain no new commands, and how the Enhanced Processor & Integrated Communications Card (EPIC) that can provide an Ethernet capability. This interface could enable a new crew interface to monitor PL Timeliner execution. The new interface could operate on either a Payload Computer System (PCS) or a Station Support Computer (SSC) that is plugged into either the Payload LAN or the Operations LAN. This will make connecting to the PL MDM more flexible and increase output capacity for transmission of commands and large amounts of telemetry to the PCS.

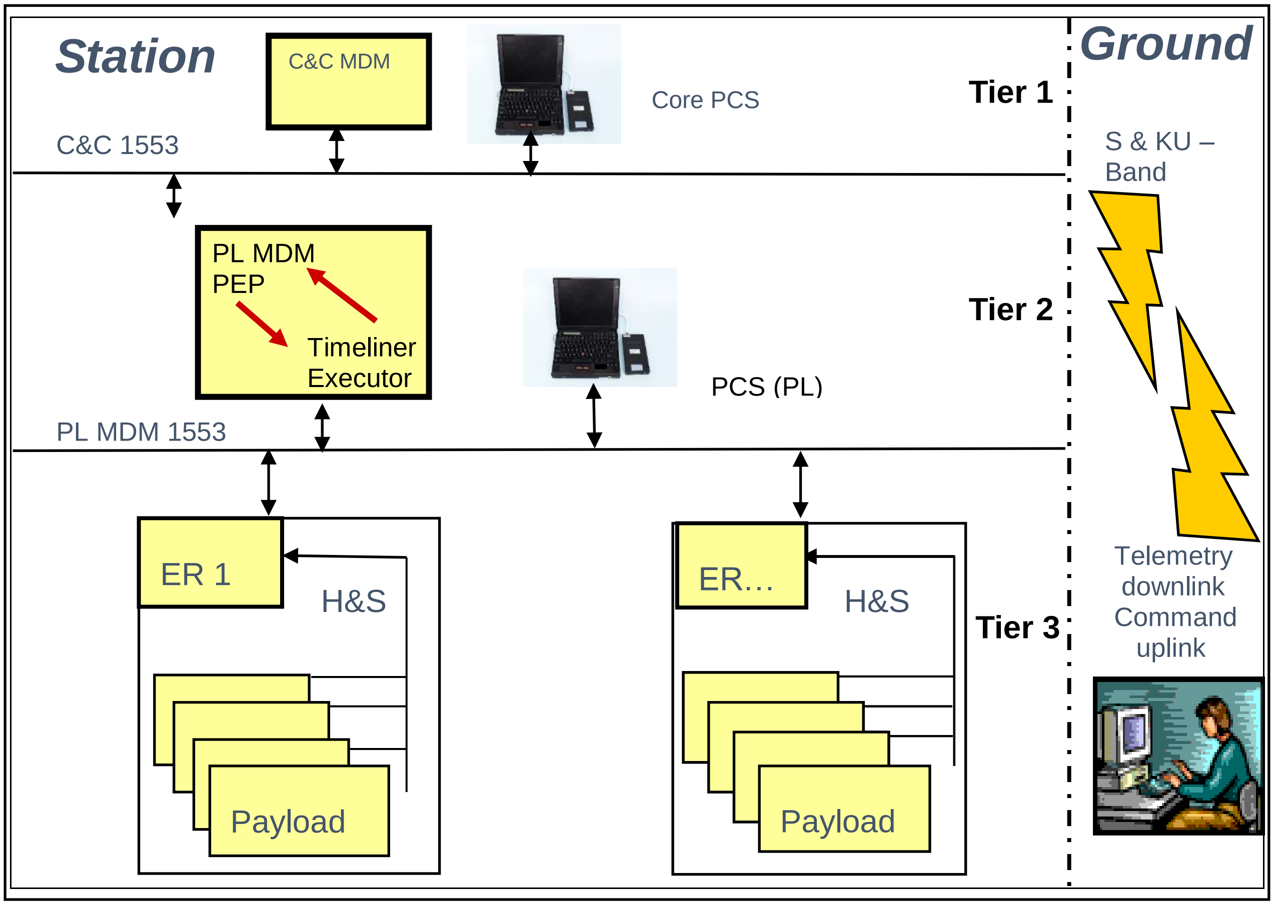


Figure 2. **ISS Tier Architecture.** *Figure 2 shows the tier architecture and the realationships between the C&C MDM, PL MDM, ISS laptops (core & payload), and experiment payloads that are installed in the US Lab on the ISS. Commands are received from the ground at the C&C MDM and will be passed to the PL MDM and onto the appropriate tier 3 RT's. Telemetry in the form of H&S and Application Process Identifier (APID) or science data*

either is passed back to the PL MDM for downlink or is downlinked directly to the ground through the KU COM unit.

II. Timeliner UIL Fundamentals and Constructs

The Timeliner UIL provides a way to programmatically model human decisions. Its constructs relate directly to the decision processes made by ground operators each day concerning such things as activating and monitoring facilities and payloads and any other procedural processes. As long as the PL MDM and the Timeliner Executor is active, there can be constant communication with the facility or payload that is to be monitored. Ground personnel are dependent on satellite communication and ground equipment to receive telemetry and issue commands. Timeliner has English language constructs that allow someone with little programmer experience to not only follow and understand execution, but to write scripts.

Timeliner scripts are hierarchically organized. The top level construct, the *bundle*, is a place holder for *sequences*. Within each bundle there has to be at least one sequence, which is the executable code that can start executing autonomously or by ground commanding. See Figure 3. The PL MDM can have 50 bundles installed into memory at one time and within each bundle there can be a maximum of 10 sequences. The core side executing on the C&C MDM can have 10 bundles with 50 sequences each. A bundle is loaded into memory on the PL MDM and the sequences within the bundle will contain all the executable scripting logic and commands. Below is an example of the structure of a bundle that contains two separate executable sequences. See Figure 3. Anything after the “--“ symbol on a given line will not be executed and is used as comments to help document the logical flow of the execution statements.

```
TIMELINER_BUNDLE - - BUNDLE 1 BEGINNING

     START MANUAL_SEQUENCE 1 - - SEQUENCE 1 START

          LOGIC - - START WHENEVER, EVERY, LOOP, IF

          COMMANDS - - COMMANDS EMBEDDED INTO THE LOGIC

          LOGIC - - END WHENEVER, EVERY, LOOP, IF

     END MANUAL_SEQUENCE 1 - - SEQUENCE 1 END

     START AUTO_SEQUENCE 2 ACTIVE - - SEQUENCE 2 START

          LOGIC - - START WHENEVER, EVERY, LOOP, IF

          COMMANDS - - COMMANDS EMBEDDED INTO THE LOGIC

          LOGIC - - END WHENEVER, EVERY, LOOP, IF

     END AUTO_SEQUENCE 2 - - SEQUENCE 2 END

END TIMELINER_BUNDLE - - BUNDLE 1 ENDING
```

Figure 3. **Timeliner code structure**. *This figure demonstrates the Timeliner Code Structure and how sequences are contained in a single bundle. The example bundle contains 2 sequences. The first sequence will have to be*

commanded from the ground or onboard to start execution and the second sequence will start executing as soon as the bundle is loaded into memory on the PL MDM. There has to be at least one sequence in a bundle and a max of 10.

III. Initial Timeliner Operations

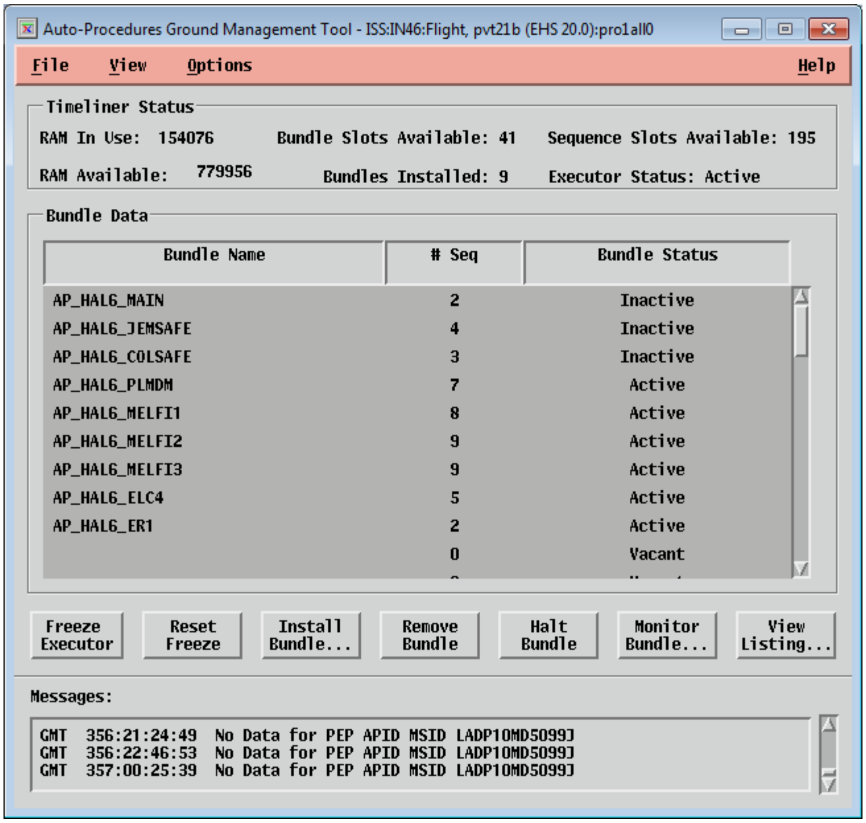
While Timeliner had been developed with spacecraft applications in mind, up to this point, it had not been fully tested on the space qualified processors that would fly on the ISS. The first step toward implementing Timeliner applications on ISS was to get the command interface working. The ISS architecture is designed on a tier structure of MDMs (see Figure 2). The MDM(s) are linked together by an IEEE 1553 bus architecture. While this architecture is robust for command and control, it does present problems when dissimilar processors are linked together through the interface, such as the way Motorola and Intel processors are used on ISS. The word order of telemetry is different for these processors and creates mapping problems when passing telemetry from one processor to another. Words are represented in a format referred to as big-endian for Motorolla or little-endian for Intel processors. With big-endian the most significant byte of a word is stored at a particular memory address and the subsequent bytes are stored in the following higher memory addresses, the least significant byte thus being stored at the highest memory address. Little-endian format reverses the order and stores the least-significant byte at the lower memory address with the most significant byte being stored at the highest memory address. Little-endian byte order is used by most MDM's on ISS while big-endian is used in some payloads that contain microprocessors such as the cryogenic freezer GLACIER.

Timeliner does not handle the formatting differences well and some data manipulation has to occur before the telemetry can be processes. Because the Timelienr UIL is developed using the Ada 95 standard and does not return a remainder when using the MOD function to swap a signed integer value, it can't be used for word swapping. When using the MOD function for a signed integer, the higher order bit will be dropped and the result will be an undesirable value. This can be overcome by declaring a data base value that maps to the same memory address as the unsigned integer value. This will result in being able to use the higher order bit as a value instead of the sign bit and allows for proper word swapping within the TIMELINER Executor. This is further complicated with byte swapping of data at each 1553 Bus Interface Adapter (BIA). In order to uplink the Timeliner bundles to the PL MDM, they must be byte swapped on the ground before uplinking to the destination processor on ISS. This creates several problems in a tier structure of more than one level. As data is propagated through processors, byte swaps occur depending on the number (odd or even) of similar MDM processor cards that it passes through. This challenging hurdle was overcome by closely working with the various Interface Definition Documents that describe the ISS command and control structure, as well as through analysis and testing.

One of the primary safety constraints placed on payload Timeliner bundles is that they can only use commands that are fully instantiated and embedded in the executable. This means that the bundles can only send predefined and fully tested commands, rather than allowing Timeliner to create updateable commands based on variable components. This method made it very straightforward to verify that hazardous commands can't be sent inadvertently from the PL MDM by a Timeliner sequence [3]. Command field mnemonics can contain modifiable and updatable fields and all the modifiable fields have to contain hard coded values in the Sequence scripts at compile time so they can't be modified by the scripts as they execute.

Timeliner commands are processed on the PL MDM like commands that are sent from the ground. When the PL MDM Payload Executive Processor (PEP) software, which is basically its operating system, receives a command from Timeliner, it adds the time stamp and checksum and then processes the command exactly like a ground command. After the checksum is added, PEP will then route the command to the appropriate bus and RT destination. See Figure 2.

The first Timeliner bundles on the ISS were simple and their sequences were executed, on command, by ground operators using the Auto-Procedures Ground Management Tool (APGMT). See Figure 4. The first operational bundle was in support of the payload Microgravity Acceleration Measurement System (MAMS). See Figure 5. This experiment consists of accelerometers housed within EXPRESS Rack 1 in the US Lab. This payload has the requirement, when active, to downlink a cyclic memory buffer to the ground every 24 hours. Initially, ground controllers performed the series of 3 commands daily. After the bundle was uplinked, ground operators could issue a single command that started the MAMS_DAILY_DOWNLINK bundle on the PL MDM that would downlink the data twice and then reset the buffer. While not completely autonomous, it paved the way for more onboard automation in support of ISS science operations.



RAM In Use: 154076

Bundle Slots Available: 41

Sequence Slots Available: 195

RAM Available: 779956

Bundles Installed: 9

Executor Status: Active

Bundle Name	# Seq	Bundle Status
AP_HAL6_MAIN	2	Inactive
AP_HAL6_JEMSAFE	4	Inactive
AP_HAL6_COLSAFE	3	Inactive
AP_HAL6_PLMDM	7	Active
AP_HAL6_MELFI1	8	Active
AP_HAL6_MELFI2	9	Active
AP_HAL6_MELFI3	9	Active
AP_HAL6_ELC4	5	Active
AP_HAL6_ER1	2	Active
	0	Vacant

Messages:

```
GMT  356:21:24:49  No Data for PEP APID MSID LADP10MD5099J
GMT  356:22:46:53  No Data for PEP APID MSID LADP10MD5099J
GMT  357:00:25:39  No Data for PEP APID MSID LADP10MD5099J
```

Figure 4. **Automated Procedure Ground Management Tool (APGMT).** *This tool is the Graphical User Interface the PRO uses to operate and maintain Timeliner bundles on ISS. The GUI shows 9 bundles that are loaded into memory on the PL MDM. In order to see sequences in a particular bundle, the bundle has to be highlighted and the "Monitor Bundle" push button initiated.*

The next logical step was to be able to read and interpret onboard telemetry. To do this, Measurement Stimulation Identifier (MSID) addresses must be shared or known to the Timeliner compiler and a address map file had to be generated at compile time. This map file defines the MSID start address, start bit and length to the Timeliner executor on the PL MDM. To verify these addresses was a huge task. There was not just one group of users who owned these addresses and in the end, a new process had to be developed to get these addresses. The byte swap issue and Intel-Motorola memory representation was a problem, and is still an issue developers have to overcome even today. But these can be overcome with testing and understanding of where the data is coming from, and how many MDM bus interface adapters the data is going through. See Figure 2.

Once the telemetry issues were understood, more complex bundles could be created. But the developers also realized that having many stand-alone, complex Timeliner bundles operating on a single MDM would soon become difficult to manage from the ground. Thus, a system was needed onboard to manage these bundles and the Higher Active Logic (HAL) concept began to take shape [3].

IV. Higher Active Logic (HAL) System

The concept for HAL was driven by constraints of bundle management, processor utilization limits, and crew, vehicle, system, and payload safety. The architecture of the ISS command and control system is based on locations, devices or Remote Terminals (RTs). Each RT location has a Remote Power Controller (RPC) that determines if the unit is powered. The first fully autonomous sequence (RT_MONITOR) was created to monitor the status of these RPCs (powered or unpowered) and turned out to be a very reliable and safe method to manage which specific payload bundle should be installed and which sequences in those bundles should be autonomously activated or not. If a particular payload RPC is “open” or unpowered, the active sequences in that bundle are halted and the associated bundle for that RT is removed from memory. This enhances safety concerns of inadvertent commanding by removing the command sources. Conversely if the RPC is “closed” or powered, the bundle is installed. If a sequence in the installed bundle should be started immediately, it will be set to go active when its bundle is installed, thus making it an autonomously activated sequence. If the bundle contains sequences that are not desired to start immediately, the PRO will have to command it to start when appropriate. With this control mechanism in place, autonomous monitoring or commanding is only possible when a RT, or payload, is powered and the desired level of monitoring or commanding is well controlled. This addressed one of the primary safety concerns of the ISS program managers.

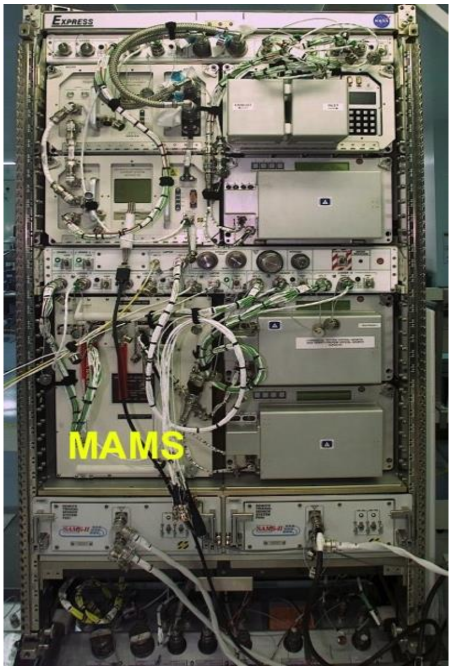


Figure 5. **EXPRESS Rack 1.** *This figure shows the EXPRESS Rack 1 that contains the MAMS Payload which the first Timeliner sequence commanded too.*

HAL was assigned 1288 words in a contiguous memory location on the PL MDM to use for shared values and messages. This gave Timeliner the ability to read telemetry, send commands and then create and update HAL defined memory. The HAL memory was mapped exactly like all telemetry and is divided into Payload Unique Identifiers (PUIs) and assigned addresses, start bit, length, and data type. The memory assignments consisted of PUIs containing 60 characters of ASCII for status, 16 bit integers and 32 bit integers that are downlinked to the ground at a 1 Hz rate (once a second). The most important aspect of HAL memory is that it can be used to communicate with other Timeliner sequences using their own telementry and messages, or for ground acknowledgement or instruction. This enabled the HAL bundles to inter-communicate with RT level bundles, share memory, and downlink statuses to ground operators.

By using shared memory, HAL is designed to actually monitor bundle sequences, execute and report back if the system is operating nominal or off nominal, and take corrective action or terminate processing when necessary. Also, this method of communication can be used to request other services from HAL by designating service request words. The important thing to note is that specific memory for HAL is being used and maintained by the HAL system and is downlinked to the ground. This was a first in space operations as actual text can now be updated and sent to the ground. This enables ground operators to view messages directly instead of using extensive look up tables that are numerous and difficult to maintain [3].

The HAL system aids in automatically configuring the PL MDM and monitors RT(s) for power, Health and Status (H&S), and sends appropriate commands. The ground operator will install HAL_MAIN when the PL MDM is fully functional and the PL MDM and the HAL system will be configured appropriately. When the PL MDM PEP software is upgraded at the end of 2016, part of the PL MDM initialization process will be to install HAL_MAIN, making ground commanding to load it obsolete. When installed, the HAL system continuously monitors power and H&S from all facility RT(s) and some subordinate payloads. The new PEP software load will make the HAL system fully autonomous and provides a fully automated recovery when the primary PLMDM is transitioned to the backup.

V. Timeliner Approval Processes for ISS Operations

Timeliner bundle development initially consists of making sure all telemetry and commands necessary for creating the bundle and sequences are available in the data base to the Timeliner compiler. Once this has been completed and the code written, the first compilation(s) can take place. The Timeliner UIL identifies compilation errors as a “cuss” or “cusses.” After all the “cusses” have been resolved and a clean compilation is made, all of the supporting files are automatically created by the Timeliner compiler. These files are the address map file ”bundle_name.tla” and the executable ”bundle_name.tlx.” These must be placed on the PLMDM Solid State Mass Memory Unit (SSMMU) for execution. The Timeliner Executor resides on the PL MDM and reads these files into memory for execution when a bundle is commanded to be installed into memory.

The ISS simulation facilities that have a Functional Equivalent Unit (FEU) PL MDM are limited resources. Some examples are the Payload System Integration and Verification Facility, Payload Rack Check Unit and Space Station Training Facility located at MSFC. These facilities and others must be scheduled in advance for testing. The scheduling and availability of these resources are a large factor in the time needed for testing. After development and testing are completed, a comprehensive review and approval process is in place prior to uplinking and executing the new script on ISS. This process involves multiple organizations including the Automated Procedures Control Panel (APCP) and Payload Operations Integration Facility Management (POIF MGT) at MSFC and the Payloads Software Control Panel (PSCP) and the Avionics and Software Control Board (ASCB) at Johnson Space Center that ensure all parties involved are aware of and concur with the new executables. The approval process typically takes about 8 weeks, but has been accomplished in 6 weeks.

VI. Autonomous ISS Sequences

For payloads, there have only been two non-HAL and several HAL autonomous sequences that have operated and are currently operating on ISS. To date, there have been two significant autonomous sequences developed and executed on ISS that were not part of the HAL system. Section A will discuss those 2 sequences, and the other sections will concentrate on autonomous HAL contributions.

A. AMO ER7 Core and Payload Sequences

The AMO experiment consisted of 2 bundles and 9 sequences that were each executed by ground commanding and as fully autonomous sequences being initiated by one command to start execution. The AMO_PLSS bundle contains all ISS core side command responsibilities and the AMO_RACK bundle contains all ER7 command responsibilities. The AMO_PLSS bundle contains 4 core element command sequences that are used to establish water flow for cooling, close main and auxillary controllers for powering ER7, and enabled smoke detection for ER7. The AMO_PLSS bundle contains sequences Act_ER7, Thermal_Flow, Power_On, FDS_Enable, and ER7_Deact sequences. The AMO_RACK bundle contains 4 sequences that will monitor ER7 as it initializes and perform telemetry checks on it to decide if it is in a proper configuration or if it should be manually configured to nominal, and after it is properly configure, the Close_Out sequence will downlink internal ER7 memory to displays that the PRO will print and place in books for historical reference.

B. HAL Main Sequence

HAL_MAIN will be loaded into memory by ground operators anytime a PL MDM is initialized or transitioned. HAL_MAIN will load the JEM_SAFE, COL_SAFE, and the PL MDM bundles. The PL_MDM bundle will automatically start execution of the PL_MDM_INIT sequence that configures the PL MDM by enabling all the profiles for downlink (APID stream 958 HAL Memory, streams 1071, 1072, and 1073 for Timeliner Satus, stream 1074 Broadcast Ancillary Data, stream 1075 Payload Ancillary Data, and stream 1076 File Memory Transfer), starts lateral transfer to the COL Lab, and starts the RT_MONITOR sequence. Starting late this year, PEP R12 will automatically load HAL Main into memory as part of the process when a PL MDM is commanded to operational mode and will make the HAL system fully autonomous. There are currently 6 fully autonomous and continuously running sequences within the HAL system. Each fully autonomous sequence is discussed below.

C. HAL RT_MONITOR Sequence

The RT_MONITOR sequence will loop every 3 seconds to check all the ISS RPC's that power NASA facility racks to load their respective bundles and monitor H&S of specific payloads that require bundles to be installed. If the RPC is closed (ON), it will start the input/output I/O function on the PL MDM and load any required bundles into memory. The I/O function allows commands to be sent from the PL MDM to the resepective facility or payload. If the H&S is inactive for payloads with installed bundles or facility RPC's are open (OFF), RT_MONITOR will stop the I/O function on the PL MDM and remove any installed bundles. This will insure that there are no bundles installed into PL MDM memory that are not needed. If a bundle has an active sequence, the sequence will be halted and then its bundle will be removed.

D. HAL AUTO MAMS DOWNLINK Sequence

The auto MAMS downlink sequence is housed in the ER1 bundle which is loaded into PL MDM memory when Health and Status (H&S) is received for the MAMS payload on the PL MDM and will automatically start executing the MAMS_AUTO_DOWNLINK sequence. The MAMS bundle will monitor for GMT 18:00 and at 18:00, it will check for Automated Payload Switch (APS) throughput from ER1 to the KU COM unit. If a good status is received, the sequence will downlink the MAMS buffer twice and then reset it to the Point of Beginning (POB). If throughput is not available, it will loop every 5 minutes and check until the status is good or the sequence is stopped by ground commanding. After the MAMS buffer has been successfully downlinked to the ground, the sequence will monitor for GMT 18:00 and repeat the check/downlink process until it is stopped by ground commanding or a PL MDM error occurs.

E. HAL EXPRESS Logistics Carrier MONITOR Sequences

There are currently 4 EXPRESS Logistics Carrier (ELC) Timeliner bundles that contain 1 sequence each that will monitor the status of its respective ELC Pre-Heaters (Channel 14 & 15) ever 5 minutes for a heater power on state. The sequences will activate when the bundles are installed into PL MDM memory to power off the heaters and prevent excessive heat from building up in the ELCx EXPRESS Pallet Adapter (ExPA) should the heaters be turned on. The ELC Low Voltage Power Supply (LVPS) Auxiliary DC/DC Converter is required to be maintained at a temperature below 25 degrees Celsius to prevent damage to the tantalum capacitor and if the heaters turn on, it can exceed this temperature limit. The heater channels can be powered on due to a power cycle of the Main and/or Auxiliary RPCs that power the ELCs and the Timeliner bundle will power off the ELC Pre-Heaters to prevent the LVPS Auxiliary DC/DC converter from exceeding the 25 degree Celsius limit.

F. AMO EXPRESS Rack 7 Sequences

There are currently 2 Autonomous Mission Operations (AMO) ER 7 bundles that are not part of the HAL system that have operated on ISS multiple times and have been used to incorporate a one button activation and a one button deactivation of the ER7 facility.

The two bundles consist of core command and rack command capabilities. The core consists of Thermal Flow, Power Controllers, and Smoke Detection that enable the rack to operate and the facility or rack bundle monitors the Rack Interface Controller (RIC), subsystem valve, Solid State Power Control Module, payload lockers, Payload Ethernet Hub Bridge and Local Area Network to insure the rack is properly configured. Each sequence in the bundles model either the JSC SPARTAN (responsible for ISS power systems) or ETHOS (responsible for ISS thermal systems) or MSFC PRO (responsible for ISS facilities and payload operations) console position operations and ground command procedures for EXPRESS 7. When activated by ground controllers, more than 60 commands can be sent to properly configure an ER. The activation of an ER is the most complex of activities performed by PROs, and the AMO experiment demonstrated that properly written scripts and ground operators can work in unison to create a more reliable and efficient way to operate than has ever been done on the past [2]. This experiment will be followed by a second experiment, in which ISS crew will both initiate and monitor the progress of activities onboard ISS. This task will be performed using NASA developed software running on an ISS computer that monitors the start and completion of bundles, and displays the step completion status to the crew.

VII. Non-Autonomous ISS Sequences

All other bundles used in the HAL system use sequences that are started and stopped by ground operators. HAL will load the bundles into memory when a facility is powered or a payload is sending H&S, however, sequences will remain dormant until a ground operator commands sequence execution.

G. HAL JEM and COL Safeing Sequences

The first bundles HAL main loads into memory on the PL MDM when installed are the JEM and COL safing bundles. In an emergency situation such as a fire or loss of cabin pressure, the JEM or COL safing sequences can be run, by ground command only, to remove power from all NASA owned facilities in the effected module. Neither of these sequences have ever been executed other than in a test environment and hopefully never will be.

H. HAL Minus Eighty Degree Freezer (MELFI) Sequences

There are currently four HAL MELFI bundles that can be run to control or monitor the MELFI facility. There is one bundle for each MELFI facility that resides on the ISS. The sequences in each bundle contain identical logic, but will send commands appropriate for the facility it is initializing or monitoring. Sequences are used to configure the initialization of the facility, configure the dewar, start or stop the Brayton machine which cools MELFI dewars, and monitor temperatures in dewar 4 when it is set to a +4c temperature. The dewar 4 monitor sequence is the most used sequence and it monitors for an over cold condition and turns the dewar off to prevent sample degradation. When the dewar cools to less than or equal to 1c temperature the sequence will power off the dewar and monitor for it to warm up to 2c at which time it will repower the dewar.

I. HAL Robotic Refueling Mission (RRM) Sequence

The ELC4 bundles RRM_SWAP sequence is commanded to execute by ground commanders. Upon initialization it will cyclically disable ELC4 discrete channel 6 for 5 minutes and 30 seconds and then will enable discrete channel 6 for 30 seconds so RRM can see all of their telemetry. RRM has 10 signals to report as telemetry for the ELC4 analog lines. If discrete 6 is disabled, Telemetry set A is active and if discrete 6 is enabled, telemetry set B will be active. Telemetry set “A” represents pressures within the payload and telemetry set “B” represents temperatures. Basically the sequence was created to compensate for a planning issue concerning bandwidth allocation and demonstrates the flexibility of uses for Timeliner sequences. This would be an impossible task to perform from the ground when you consider unexpected Loss Of Signal (LOS) periods with no command or uplink capability.

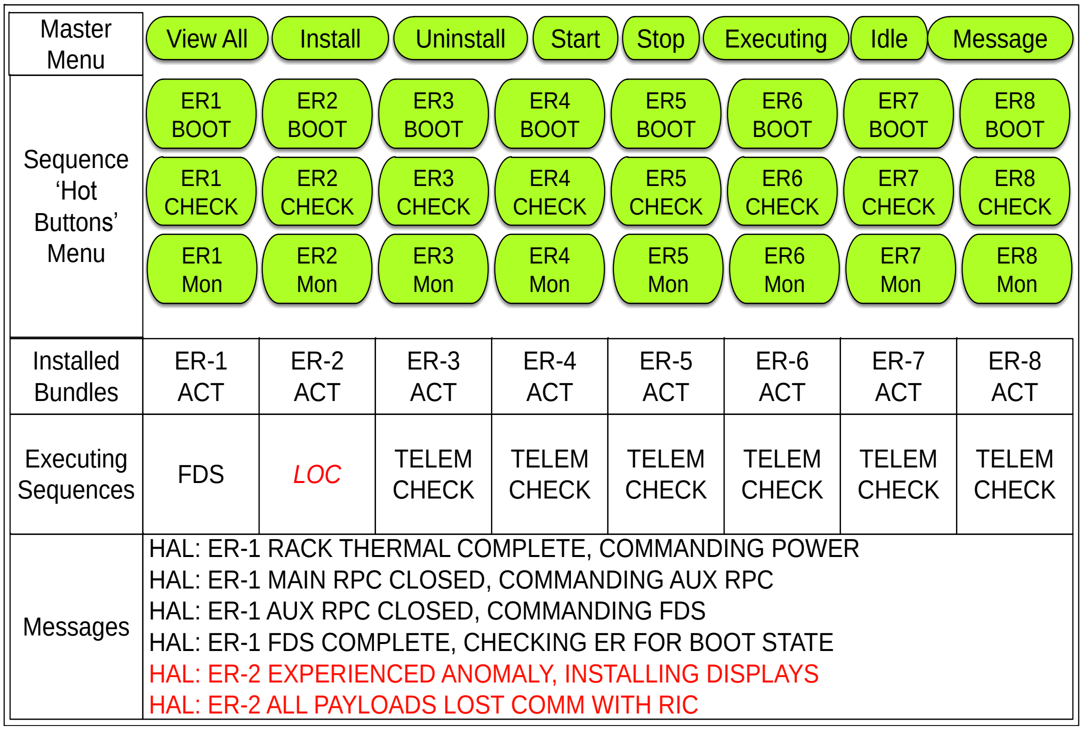
VIII. Future Autonomous Systems on ISS and in Control Centers

The HAL system has worked well and proven that operator knowledge can be reliably programmed into executable sequences that operate on a vehicle. As HAL functionality has grown, the inefficiencies and cumbersomeness of operating it with the current tool set is becoming more apparent. In order to fully utilize the knowledge gained and the systems available, more sequences and a more complex tool has to be developed. Currently for PRO to activate an ER, they have to manually initialize 4 displays and open as many as 3 different books to have all the tools available they may need. This cumbersome and complex routine can be automated with a new user interface (UI) and more robust sequences that will interact with the UI. The new system, HALO, is discussed in more detail below.

J. Higher Active Logic Organizer

The current HAL bundles used on ISS serve as a good starting point for future applications that will help prove the systems value to manned mission operations. However, the current system lacks features and benefits to operations that can be obtained by more robust scripts and a enhanced interface communicating with the scripts. The current system allows for only one sequence to be viewed at a time and there are no AMO like bundles in HAL. As mentioned earlier, there are several autonomous bundles that contain sequences that automatically start when there bundles are loaded into memory. Adding more bundles with autonomous sequences that can monitor the health of more facilities and payloads, as well as the status of HAL itself, is vital to the success of long duration missions and future laboratories that are placed in Low Earth Orbit (LEO) and into outer space. The Higher Active Logic Organizer (HALO) is a MSFC ground tool that will control and interface with the bundles and sequences that reside on ISS. The HALO concept will require two distinct components that can communicate with each other. For ISS, the operator interface will reside on the ground in servers used for flight controllers. For deep space and habitat applications, the operator interface will reside on the vehicle as a GUI. The second component will reside on the PL MDM or a PL MDM like computer as scripts executing within the HAL system and communicating to the HALO

GUI. The GUI can be located on the ground or within the vehicle, The computers used for deep space and the orbiting habitat have not been determined, but will need to contain a Timeliner like functionality.



Master Menu	View All	Install	Uninstall	Start	Stop	Executing	Idle	Message
Sequence 'Hot Buttons' Menu	ER1 BOOT	ER2 BOOT	ER3 BOOT	ER4 BOOT	ER5 BOOT	ER6 BOOT	ER7 BOOT	ER8 BOOT
	ER1 CHECK	ER2 CHECK	ER3 CHECK	ER4 CHECK	ER5 CHECK	ER6 CHECK	ER7 CHECK	ER8 CHECK
	ER1 Mon	ER2 Mon	ER3 Mon	ER4 Mon	ER5 Mon	ER6 Mon	ER7 Mon	ER8 Mon
Installed Bundles	ER-1 ACT	ER-2 ACT	ER-3 ACT	ER-4 ACT	ER-5 ACT	ER-6 ACT	ER-7 ACT	ER-8 ACT
Executing Sequences	FDS	*LOC*	TELEM CHECK	TELEM CHECK	TELEM CHECK	TELEM CHECK	TELEM CHECK	TELEM CHECK
Messages	HAL: ER-1 RACK THERMAL COMPLETE, COMMANDING POWER HAL: ER-1 MAIN RPC CLOSED, COMMANDING AUX RPC HAL: ER-1 AUX RPC CLOSED, COMMANDING FDS HAL: ER-1 FDS COMPLETE, CHECKING ER FOR BOOT STATE HAL: ER-2 EXPERIENCED ANOMALY, INSTALLING DISPLAYS HAL: ER-2 ALL PAYLOADS LOST COMM WITH RIC							

Figure 6. Higher Active Logic Organizer (HALO) GUI. *The User Interface (UI) above is a conceptual first cut at what HALO may look like. The eventual UI will mature and look much different..*

K. HALO Ground Tool

The Higher Active Logic Organizer (HALO) ground tool will control and monitor all activities associate with the HAL system on ISS. It will be able to issue ground commands to install and uninstall bundles on the PL MDM, monitor the heath of facilities and payloads on ISS, report to ground operator's issues that occur on ISS and in some instances take corrective action. A conceptual view of what the tool will look like can be seen in figure 6. The top menu bar options are used to view all bundles available, install/uninstall bundles, start/stop sequences, view all sequences that are executing, view all sequences that are not executing, and configure the message area. There are currently 8 NASA owned EXPRESS Racks on the ISS and there will be "hot" buttons available in HALO for more complex sequences such as activating, deactivating, telemetry checks, and performing a MCC reboot on each EXPRESS Rack. Hot buttons should be used for sequences that will be used a lot or sequences that are complex in nature and would otherwise require a ground operator to start several sequences for a single operation. The message area will display messages from PL MDM HAL memory and messages generated from the sequences executing in PL MDM memory. The message area can be configurable based on individual user needs and desires. In addition to controlling sequences operating on the PL MDM, HALO will be used to provide procedures and displays to ground controllers so they can react appropriately to events occurring on the ISS. The bundle and sequence structure on the PL MDM will provide status from systems on ISS which will be discussed in greater detail later.

L. HALO Display Architecture

The HALO ground tool will not only manipulate bundles and sequences on ISS in the HAL system, but it will act as a control tool to aid the ground operator in decision making and provide them with all the necessary tools they will need to accomplish specific tasks. HALO will communicate with HAL sequences which are constantly monitoring Facility and payload telemetry on ISS and will alert operators about anomalous conditions occurring within the ISS facilities, payloads and HAL. When HALO recognizes an anomaly has occurred on ISS, it will display a message in the message area and then either initialize or pop to the forefront any command or telemetry displays the ground operators will need to resolve the anomaly. In addition to command and telemetry displays, HALO will initialize the appropriate command procedures ground operators will need to follow while resolving the anomaly. By doing this, HALO will in effect become another ground operator. This will help to greatly reduce human operator stress and lessen the probability of human errors. Capability similar to this has been demonstrated in a medium fidelity human space mission simulation [5]; the set of off-nominal conditions is programmatically mapped to either display pop-ups, fault response procedures, or both.

An example of the interaction between the HAL and HALO systems during an anomaly would be when an ER's Rack Interface Controller (RIC) locks up and the EXPRESS Rack can't transmit any of its payloads Health & Status (H&S) to the PL MDM. This could be a serious condition, especially, when one of the payloads has a Caution & Warning (C&W) word in its H&S sent to the PL MDM. The C&W word is interpreted by the PL MDM and can be used by the C&W tables to send a command to power off a payload when a C&W temperature limit is exceeded. If the PL MDM is not receiving H&S from the RIC, it doesn't have the ability to monitor the payloads C&W word. By recognizing when the RIC has locked up and for HALO to place everything the PRO will need in front of him is a good step in the direction toward fully autonomous systems needed for future operations.

M. HAL Bundle and Sequence interaction with HALO

The HALO tool will be capable of installing/uninstalling, starting/stopping, and monitor bundles and sequences loaded onto the Solid State Mass Memory Unit (SSMMU) on the PL MDM. Bundles supporting HALO will be organized into 4 separate categories (Core, EXPRESS Rack, Payload, and Monitoring) for operational support. The Core bundles will contain sequences that are responsible for commanding to all the assets that provide power, thermal, and fire detection for each facility. The rack bundles will contain all the sequences that are responsible for facility and EXPRESS rack configuration items after a facility or Rack has been powered. The payload bundles will contain sequences that operate and control payloads on the ISS, and the monitoring bundles will contain sequences that monitor facility H&S, EXPRESS Rack H&S, or payload H&S and report to ground operators the health of those systems or take corrective actions by issueing commands from the sequences. By logically organizing the bundle architecture based on the type of systems used, it will lessen the update and approval process and keep testing to a minimum.

The HAL system has proven over the last 11 years to be a very reliable system to monitor and control the execution and use of Timeliner bundles and their sequences on the ISS. The knowledge gained from the past should be leveraged for future automation to make ISS, habitats, and future space vehicles less reliant on manual procedures and ground operators. Knowledge based intelligent systems will be mandatory for missions with long communication delays in order to operate and control complex onboard systems. Timeliner allows operator knowledge to be palced within the sequences which will provide a higher level of scripting functionality. Initially, the development and test bed for creating and demonstrating HALO will be at the MSFC for ISS use. After implementation for ISS is complete, HALO could be ported to any vehicle or habitat with little modification to its kernel. There will need to be more bundle development in the areas of monitoring and controlling ISS facilities and payloads for the HALO concept to be fully implemented, but there is a good foundation in operation today and available system experts with knowledge to quickly implement a useful solution.

IX. Opening up the EPIC card on the PL MDM for a Crew interface

During the development of the AMO EXPRESS experiment [2], it became clear that an autonomous crew of a future spacecraft needs less information than the flight control team about the status of autonomous procedures. Nevertheless, the crew can benefit from a subset of the information in the HALO ground system design. For instance, the ability for crew to know at a glance that some activities have started, are ongoing, or are completed, will increase their situational awareness. This information can increase science return for payloads on ISS today, but becomes more important, both for payload oeprations and for core spacecraft systems management, in future Exploration missions.

Since the status of all Timeliner activity originates on the PL MDM, today a display concept like HALO must reside on a PCS computer on the Payload 1553 bus. This is impractical, due to limitations on the PCS software load, and the fact that PCS computers must be moved to the Payload bus, as they are not regularly deployed there. A critical ingredient to moving even a 'HALO-lite' GUI onboard the ISS is to get the PL MDM Enhanced Processor &

Integrated Communications Card (EPIC) opened up. When the PL MDM's were upgraded from a 386SX to a Pentium processor, the EPIC card was not enabled, so ethernet access isn't possible. It is currently connected but has been software firewalled by the Payload Executive Processor (PEP) Software.

In an ISS demonstration conducted in 2014 [5], software was developed and deployed on the ISS Operations LAN, and used by ISS crew on Station Support Computers (SSCs) and portable devices (e.g. iPads). The development and deploymeny was comparatively quite fast (~9 months), and the design included significant operational software reconfiguration using data files, which could be approved and effected within days. With the opening of the EPIC interface, there are large amounts of telemetry that can be transmitted to the ISS Operations LAN, and subsequently displayed on computers available to the crew anywhere onboard ISS, making the design, use and maintenance of HALO onboard ISS much simpler than fielding it on a PCS.

X. ISS AMO Tools in Development

Future crew autonomy demonstrations onboard ISS can benefit from the integration of data from disparate sources. To take one example, ISS crews exercise for several hours a day in order to maintain physical health, especially in the run up to their return to Earth. One side effect of crew exercise can be increased vibration, which must be compensated for by the ISS motion control systems. Today, flight controllers must integrate information from many different sources to determine if excess vibration or ISS motion control system activity indicates that there is a problem with the exercise equipment. Sensors on the exercise hardware reports some information that can be used to determine its state, but often this is not enough on its own to diagnose a problem. The ISS has numerous accelerometers to measure forces on ISS, as part of core systems and also in payload racks (e.g. the MAMS system discussed earlier in this paper). These sources of information, plus rate information from ISS, can be used to determine whether exercise equipment is the root cause of unexpected vibration or forces. The centralization of all of these sources of information for processing and display for the crew will require integration of data from the PL MDM, C&C MDM, and the exercise equipment. With the opening of the EPIC interface, as described above, it will be possible for ISS crew to diagnose and correct problems with the exercise equipment or other items normally diagnosed from the ground.

XI. Conclusion

Science operations onboard ISS to this point have been tremendously successful. The implementation of autonomous systems, such as HAL, have been a particularly noteworthy accomplishment. Like any endeavor, there have been setbacks, and progress has not come as quickly as first expected or hope for. However, reflecting on more than eleven years of Timeliner supported ISS payload operations; it is easy to recognize that great strides have been made with respect to reducing the workload of ground controllers and increasing operational reliability for effected systems. Specifically, in the area of autonomous systems onboard crewed spacecraft, Timeliner and HAL has dramatically changed the way these types of systems are utilized by providing a way to model human decisions and processes.

Prior to the ISS program, automation within critical systems was minimized in an effort to increase a perceived reliability and to maintain more human control. As discussed in this paper, the first payload Timeliner bundles proved that operational reliability could be increased and errors incurred from complacency, fatigue, and forgotten knowledge can be significantly decreased. And thus, they paved the way for more complex bundles, such as the HAL concept and autonomously operating sequences. The testing process and the availability of real systems to test with insure a lesser probability or completely eliminate any coding related errors. After testing and packaging future HAL Timeliner releases, there is a 6-9 week approval process for the HAL concept which allows time for more thought about the development and testing by MSFC and JSC experts to ensure the design is safe, well tested, and ensure operational procedures are followed. Now that autonomous systems have been well proven onboard on a crewed spacecraft with commanding to tier 2 and tier 3 MDM's and RT's, the natural progression would be to increase all levels of automated intelligence and to eventually design and implement a fully automated real-time monitoring, commanding, planning and re-planning system. If the vehicle is designed with consideration of the operational automated system design, more can be accomplished safely and efficiently with less crew. As we move out of LEO and further out into space, the automated system and vehicle design become more critical.

With the Space Launch System (SLS) and the Habitat structures being planned for future exploration, the hope is that the steady stream of operating payloads and acquired knowledge of Low Earth Orbit (LEO) and deep space operations will continue. Many of these science payloads will take advantage of autonomous operations provided by the Timeliner UIL. In turn, this automation will help earthbound scientists conduct their experiments as efficiently and as safely as possible.

Appendix A
Acronym List

AES = Advanced Exploration Systems
AFTS = Autonomous Fluid Transfer System
AMO = Autonomous Mission Operations
APCP = Auto Procedure Control Panel
APGMT = Auto Procedure Ground Management Tool
APID = Application Process Identifier
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
CMD = Command
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
MSID = Measurement Stimulation Identification
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

Acknowledgments

The authors would like to acknowledge the following groups who have contributed to the success of performing remote science onboard the International Space Station:

- The POIC operations team at MSFC who operate science payloads around the clock on ISS, and at the same time plan for future experiments

- The ground testing teams at MSFC, JSC, Ames, and KSC who work to ensure that hardware and software launched into space is the best it can be

- The astronauts and cosmonauts who live and work aboard the ISS and risk their lives in the pursuit of space exploration

- The scientists and engineers who design the experiments and use the results to improve our way of life on Earth.

References

[1] Charles Stark Draper Laboratory Timeliner Webpage: http://timeliner.draper.com

[2] Haddock, 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

[3] Stetson, 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.

[4] J. Frank, D. Iverson, C. Knight, S. Narasimhan, K. Swanson, M. Scott, M. Windrem, K. Pohlkamp, J. Mauldin, K. McGuire, H. Moses. Demonstrating Autonomous Mission Operations Onboard the International Space Station. Proceedings of the AIAA Conference on Space Operations, September 2015.

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

[6] Kortenkamp, D., Verma, V., Dalal, K.M., Bonasso, R.P., Schreckenghost, D., Wang, L., A Procedure Representation Language for Human Spaceflight Operations. Proceedings of the IEEE Aerospace Conference, 2013 Representation Language for Human Space Flight Operations. 9th International Symposium on Artificial Intelligence, Robotics, and Automation for Space, Los Angeles, CA, 2008