The NASA Constellation Program Procedure System

Robert G. Phillips[1] and Lui Wang[2]
Johnson Space Center, Houston, TX, 77058

NASA has used procedures to describe activities to be performed onboard vehicles by astronaut crew and on the ground by flight controllers since Apollo. Starting with later Space Shuttle missions and the International Space Station, NASA moved forward to electronic presentation of procedures. For the Constellation Program, another large step forward is being taken—to make procedures more interactive with the vehicle and to assist the crew in controlling the vehicle more efficiently and with less error. The overall name for the project is the Constellation Procedure Applications Software System (CxPASS). This paper describes some of the history behind this effort, the key concepts and operational paradigms that the work is based upon, and the actual products being developed to implement procedures for Constellation.

I. Introduction

THIS paper discusses the approach being taken in the NASA Constellation program for general control of manned vehicles. The work involves knowledge capture and development of procedures and automated scripts for onboard and ground control of vehicles, procedure validation and verification, and procedure execution in various operational modes and environments. While the initial work is targeted at the Orion manned vehicle, the goal is to develop a system that will be usable in all manned vehicles, and possibly unmanned ones as well.

Historically, a procedure in the manned space flight program is simply a set of instructions to astronaut crewmembers and/or flight controllers on the ground for performing a defined task. Procedures can vary quite a bit in structure: some are very short and others very long; some are very linear—do "A" then "B" then "C"—and some contain many pathways, notably procedures for malfunction diagnosis and corrective action. All manned space flight procedures, both for use on vehicles and in the Mission Control Center (MCC) were printed on paper for most of NASA's history.

In the early 1990's the Shuttle program started migrating non-critical procedures away from paper to PDF-based procedures viewable on non-critical computers onboard the Shuttle and in the MCC. Shortly thereafter, a PDF-based procedure viewing system was developed for the International Space Station (ISS). Both of these systems were intended to preserve the "paper procedure experience", with electronic procedures having the same appearance as paper procedures, and many similar annotation capabilities. For example, an astronaut crewmember could mark a page with a virtual paper clip for quick reference later.

At the same time that this evolution of procedure viewing was happening, a similar evolution occurred in vehicle control (see Figure 1). The earliest control systems used for manned space flight consisted of special-purpose hardware for performing actions, both on the vehicles and in the MCC. Special purpose hardware control systems gave way over time to software-based configurable control systems. This was most noticeable in:

- The migration in the 1990's of the MCC from special-purpose hardware systems to workstations with software applications for sending commands and receiving and analyzing telemetry.
- The development of the Shuttle "Glass Cockpit" displays in early 2000.
- The development of data-driven system control interfaces for the ISS in 2001.

One thing that held true in all of this advancement was that there was no interaction between the procedure viewers and the software controlling the vehicles. This was primarily considered to be a safety issue: if a procedure directly controlled the vehicle, then it would be considered to be flight software and subject to the rigorous testing and control of all flight software. But procedures often need to be updated and refined during missions. The six-

[1] Software Engineer, Tietronix, 1331 Gemini #300, Houston TX 77058, AIAA Senior Member.
[2]Advanced Technology Group Lead, Spacecraft Software Engineering Branch, Software Robotics Simulation Division, NASA/JSC MS: ER

month plus time needed to certify flight software would have prevented this. The reluctance toward allowing automated operations also delayed the use of automated scripts onboard the ISS for over ten years.

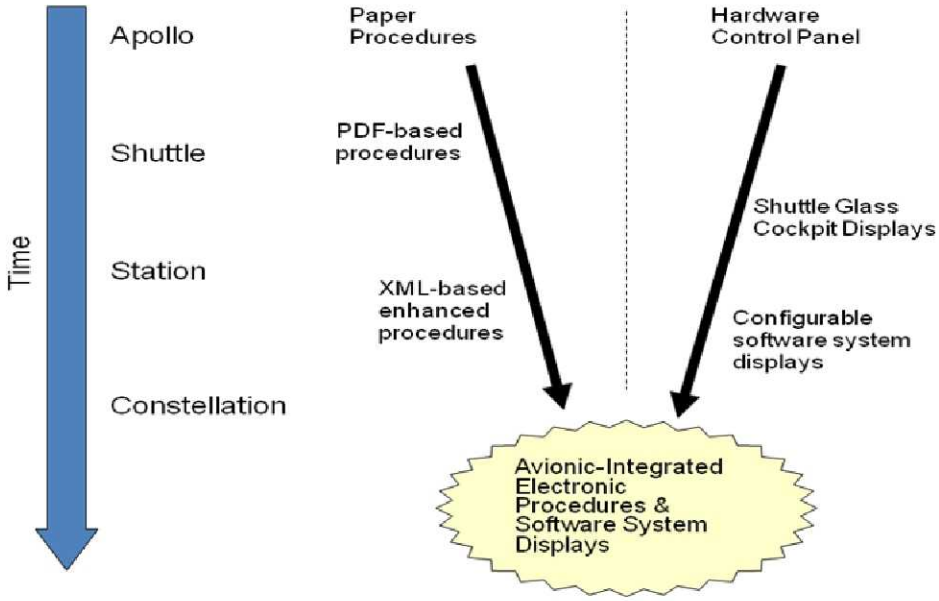

Figure 1: Evolution of Manned Space Flight Procedures and Controls

The procedure system being developed for Constellation is a definite leap forward in terms of usability and reliability. At the same time, however, it is important that the system developed be easily recognized and understood by authors and users of current procedures. There is a huge amount of experience and learning that have driven the appearance and operations of current procedures. Our goal is to build on that body of knowledge, not throw it out to start over from scratch.

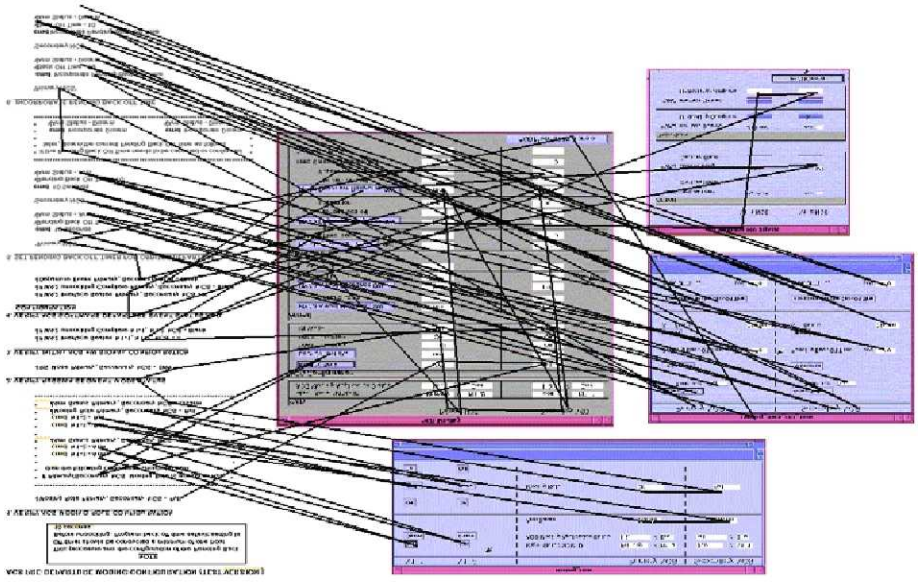

Figure 2: Early View of Static Procedures and Vehicle Displays on the International Space Station. *The user's focus has to move in a very complicated pattern between the procedure and displays.*

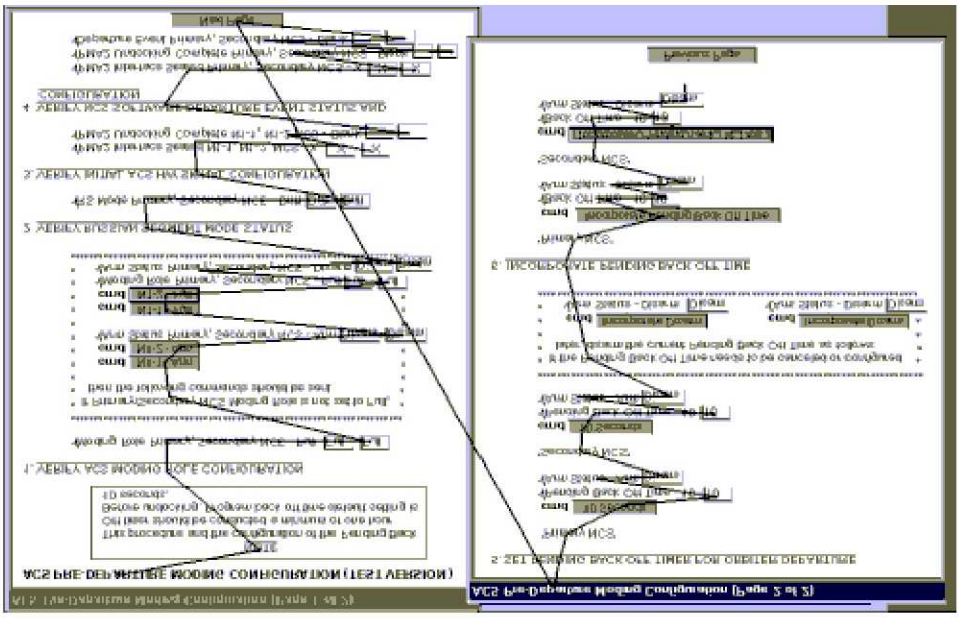

Figure 3: Early Conceptual View of Integrated Procedure System. *The user's focus is much more linear in an interactive procedure with embedded telemetry values and commanding.*

A number of new concepts in areas such as human-machine interaction, concurrent operations, dynamic planning, and automation have been explored in earlier related research projects. For example, Figure 2 and Figure 3 show some of the human factor considerations in optimizing the complex interactions between the procedures and the vehicle displays by developing the concept of Procedure Display that includes commands and telemetry feedback to linearize user's focus and minimize interactions with vehicle displays. This research has helped in the development of the Constellation procedure systems concepts. Ultimately, however, Cx continued to follow a paradigm of humans in control of manned space vehicles.

II. Key Issues

There are two key issues that have driven the conceptual design of the CxPASS and onboard Constellation procedure systems.

1. Humans in Control

The early conceptual idea shown in Figure 3 was developed in the mid-1990s. While the Integrated Procedure System was considered advancement by the authors, it never was accepted in the manned space operations community for one important reason: if a procedure can be used to interactively control a vehicle, it has to be classified as flight software. The approval process for flight software takes a minimum of six months for final certification. But procedures regularly have to be modified much more frequently than that.

a. Frequent Procedure Updates

Throughout the entire history of human space flight, procedures have been updated frequently, especially during missions. Anyone who has watched the move *Apollo 13* saw how engineers and flight controllers worked to develop new procedures for scrubbing carbon dioxide from the atmosphere and for powering up the Command Module. While the real world is seldom that dramatic, existing procedures often need to be modified or new procedures need to be created and sent to the vehicle on a very short turn-around.

b. Maintain Human Control

During procedure execution, when a procedure instruction is reached that requires interaction with a vehicle display, the appropriate action on the display is prepared, or "cued". The crewmember simply has to press the default display control to perform the action. *But the onboard executor cannot perform the action on its own.* The human crewmember must be in the loop. The crewmember can accept the default behavior, override it, or even skip ahead or back in the procedure. This will create a separation between the procedure data and the flight software, since the human is making the control decisions. This process of requiring human control allows procedures to not be classified as flight software, and will allow the frequent procedure updates necessary for human space flight.

2. Improved Quality of Execution

One of the main goals of an interactive procedure system is to improve the quality of procedure execution. This comes in three main forms.

a. Ease of Use

In earlier systems, a procedure instructs the crew on what to do, but the crew has to take action independently of the procedure. This created a problem and loss of focus between the paper-based and electronic procedures and the vehicle displays (see Figure 2). In Constellation, the procedure will provide feedback, automate navigation, and prompt actions in a more intuitive fashion. An interactive procedure will be easier to use, with fewer eye movements to track progress and fewer inputs to accomplish the instructions. As shown in Figure 2, someone executing a static procedure has to follow a very complicated sight path between the procedure and the vehicle displays. Simplifying that process and other similar parts of procedure execution makes training to use procedures and control the vehicle much easier. The procedures in Constellation will interact with the vehicle through a procedure executor instead of being static.

b. Reduction of Human Error

When the procedures are easier to use, it will not only make execution easier but will also reduce human error. Astronauts are trained extensively to avoid mistakes, but having a system that assists them by tracking progress through the procedure, checking to ensure that telemetry values are as expected, clearly reduces the chances for error and frees the user's attention for more critical items.

c. Situational Awareness

Finally, the ease of use and reduction of errors allows crew members to focus more on the big picture of the vehicle status, and to recognize and deal with off-nominal situations more quickly and surely. This improved situational awareness not only improves the quality of procedure execution, but of the entire mission. Also, the tight coupling and combined viewing of the procedure and the vehicle display allows the crewmember to maintain good situational awareness of the system being controlled (see Figure 4).

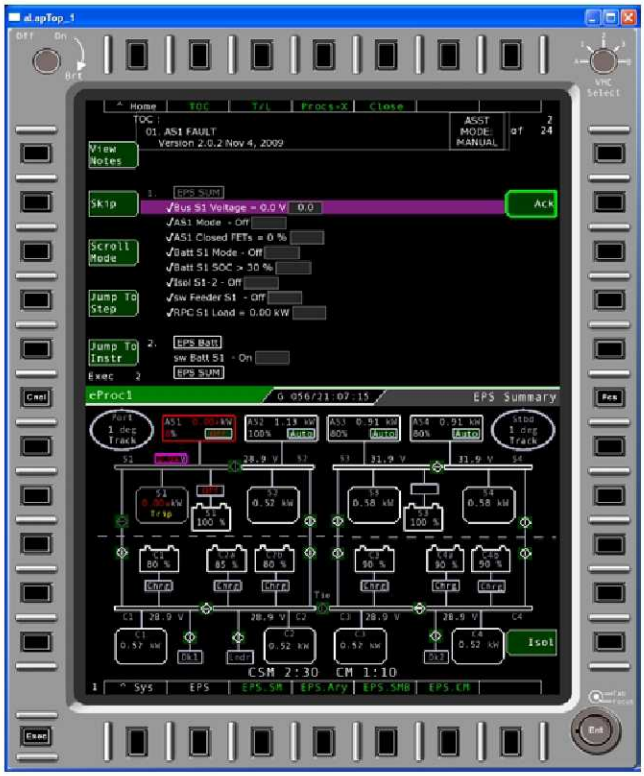

Figure 4: Proposed onboard integrated procedure display and vehicle display. *The highlighted instruction (top portion of the Display Unit) shows the current instruction step to check that Bus S1 voltage equals 0.0 with telemetry feedback box next to the instruction. The AS1 display element of EPS Summary in the vehicle display (bottom portion of the Display Unit) is boxed with red border to show the current telemetry value. If the crewmember presses the default control in the second right edge key from the upper right side with the "Ack"*

highlighted, the instruction will be acknowledged and marked as completed, and the next instruction will be selected.

III. Knowledge Capture

To create the Constellation procedure system, the first step was to understand how to capture the content of procedures. Earlier procedures with fixed rules for format and content are focused to ensure human readability using the paper-based medium. Procedure knowledge and content is captured for humans to understand, but it is not good enough for computers to perform the instructions. So, the initial activities in this project required a consolidation of how procedures will appear and operate. This was accomplished through many meetings between the procedure design group, ISS and Shuttle procedure managers, and specific groups of procedure authors and users.

In order to improve procedure authoring as well as execution, a number of key concepts are driving the design of the Procedure Authoring Tool (PAT). PAT will allow users to author procedures in various environments—in the office, in the Mission Control Center, on a laptop and so on. PAT will be designed to enable fast and easy procedure creation, with drag and drop or picklist features for telemetry, vehicle commands, and defined instruction types. PAT will use links to databases of command and telemetry definitions and vehicle display definitions to fill values such as display element identifiers and enumeration labels. These databases will also contain mappings between ground and onboard commands, and telemetry, so that procedures authored for onboard use can also be executed on the ground.

1. System Representation

Procedures will constantly reference command, telemetry, and vehicle display data. To author procedures, it will be necessary to create databases that contain all of this information, as well as all of the connections between them. For example, a vehicle display might contain a button to activate a water pump. To properly cue the command, the procedure will need to know the identifiers for the display, the element (the pump), and the command (the button). The procedure must furthermore be able to understand that sending that command is equivalent to sending a vehicle command that is used to control four different water pumps, with a parameter that specifies the pump and a parameter that specifies the "start" action. Both of these are hard-coded into the vehicle display for that button. There is also a third parameter—the pump speed—that is received from an input field on the vehicle display. Furthermore, this parameterized vehicle command could have a different ground equivalent command. All of this information is needed so that the procedure can be executed onboard, can be run as a script (if it is certified as a script), or executed on the ground. This example shows why a complete system representation is needed to create procedures.

2. Unified Authoring

A major goal of CxPASS is to implement a "write once, use anywhere" procedure system. At least eight different viewing and executing environments have been identified. It is important that the content generated when authoring procedures supports all appropriate modes of operation in all environments. The procedure data can be used to create printed procedures and automated scripts as well as executable procedures. Capturing the data to enable all of these uses in a single editing, management, and verification environment helps create consistent procedure appearance and use.

3. Consistent Content

Different flight operations groups have, prior to Constellation, independently authored their own procedures. The procedures were organized and designed as needed for their specific domains of expertise. While there have always been procedure writing standards, these have been more guidelines than hard rules and have dealt with the appearance of procedures more than the contents. In spite of these standards, many groups have developed procedures with tremendously differing styles and appearances. Some of this variation is needed to better deal with different environments, but a lot of it comes from individual styles reinforced by history and habit.

Because of the need for knowledge encapsulation and interaction, Constellation procedures will be very consistent in their structure and content. Editing, viewing, and executing applications in different environments may be able to give different views of the procedure, but the set of views will conform to a single standard in their general appearance and behavior. This is expected to improve training and reduce error in procedure execution.

The Constellation procedures will be stored using an XML schema – the Procedure Representation Language (PRL). PRL defines the kinds of procedures possible, the instructions possible in different procedures, and the required data for a procedure to be executable.

4. Explicit Functionality

A concern for CxPASS has been that current non-interactive procedures rely upon a great deal of implied knowledge and operations training. For Constellation, this knowledge must become explicitly defined and captured, either in the procedures or in database/ontological descriptions of vehicle systems.

For example, a simple command instruction "command Pump A On" implies several things – that no action is needed if pump A is already on, that the command is to be sent if it is not on, and that some unstated check is needed to confirm that the pump has turned on after the command is sent (see Figure 5). Furthermore, the same described action involves different actual computer commands and telemetry feedback if the action is executed onboard the vehicle or remotely from the ground. Currently, the crew knows through training whether to look for direct feedback (telemetry indicates pump A is on), indirect feedback (system is pressurizing), or to not wait for confirmation but instead continue. All of this information has to be captured within the seemingly simple context of turning a pump on. The user and author will see (the text in the box is live vehicle data feedback):

cmd Pump A – ON **OFF**

While the data file will contain something like the following PRL:

```
<CommandInstruction id="instr16" displayId="DSP122">
<Command commandId="ABD73512" elementId="FF182D4" groundCommandId="AF261002"
    groundCommandParameters="37.0 4 1.0">
    <Description>Pump A - ON</Description>
  </Command>
  <DesiredState>
    <CurrentValue>
      <DisplayElementValue dataId="A534CDA0" format="enumeration"
        predefinedEnumeration="enum7"/>
    </CurrentValue>
    <Equal/>
    <TargetValue>
      <IntegerValue format="enumeration"
        predefinedEnumeration="enum7">1</IntegerValue>
    </TargetValue>
  </DesiredState>
  <ConfirmedState>
    <CurrentValue>
      <DisplayElementValue dataId="A534CDA0" format="enumeration"
        predefinedEnumeration="enum7"/>
    </CurrentValue>
    <Equal/>
    <TargetValue>
      <IntegerValue format="enumeration"
        predefinedEnumeration="enum7">1</IntegerValue>
    </TargetValue>
  </ConfirmedState>
</CommandInstruction>
```

Figure 5: Example PRL XML Representation of a Command Instruction

The text "Pump A - ON" in the above example is the displayed body of the command. A requirement for the procedure data files is that all information needed to render the procedure correctly be present in the PRL content.

IV. Procedure Execution and Viewing

The Procedure Viewer/Executor (PVE) application will allow flight controllers to view in the MCC environment, and to execute procedures. It is important to note that while procedures can be written for onboard execution or for ground execution (or combined execution), flight controllers will be able to execute all procedures from the ground regardless of their intended use. A procedure written for onboard execution will be based on the onboard vehicle interfaces. The flight controllers on the ground will have different interfaces, but the same underlying operations must be performed when executing the procedure in either environment. This is an important example of why a database or ontology of vehicle systems will be vital.

PVE will have to duplicate the look and feel of procedures and execution onboard, but expand beyond that for ground-specific operations. Also, it is anticipated that scripts will be more commonly used on the ground than onboard, so support for scalable levels of autonomous execution will probably be used.

The onboard procedure executor will allow the crew to view and execute procedures onboard the Orion vehicle. As was mentioned above, the procedure execution will not directly control the vehicle, but instead prompt the crew to perform the appropriate actions. It will run on the avionics system, and share a physical screen with the vehicle display with which it interacts. Also as mentioned above, the onboard procedure data files will be optimized for onboard use.

1. Multiple Execution Environments

The Constellation procedure system will be able to support multiple execution environments, including executing procedures in multiple modes of execution. Also, it will be able to execute procedures as scripts that are intended to run with limited or no human interaction. The design also allows for the executor to provide scalable levels of procedure execution assistance to the users. The executor could, for example, evaluate as a single action a set of initial conditions that are necessary for proper procedure execution. The initial systems will only allow very limited levels of assistance, but the design allows more.

Another important aspect of this is the ability of the crew to override the executor—skipping steps, repeating steps, performing only a certain section of a procedure, and so on. Procedures are usually written with certain assumptions, and the ability of the human mind to adapt those procedures to unforeseen circumstances cannot be over-emphasized.

While this paper is about procedures, it is clear that the distinction between a script and a procedure is a matter of intended use and management, not content. Since the automated vehicle scripting capability will directly control the vehicle, it will have to go through the more rigorous flight software certification process. It is, however, reasonable to expect that a large class of procedures could be executed as scripts, if they are approved for such use. The decision to run a procedure as a script comes from the maturity and trust that a procedure can develop after repeated use. The maturation process from procedures to scripts could ease the flight software certification and has been identified as an opportunity to shorten the time to certify onboard scripts for the flight software.

All of the instructions in a procedure can be classified as ones that do or do not *require* human input. The current execution plan will require human input for each instruction, but the content of the instruction does not. The above instruction examples of commanding a pump on or verifying that a pump is on could be executed without human input provided that flight safety and reliability issues are successfully addressed. On the other hand, instructions to, e.g., close a physical valve or visually inspect a robotic arm, could not be performed without humans. Ultimately these factors define the distinction between what could or could not be automated.

2. Multiple Procedure Views

In addition to the various execution environments described above, there will be a number of viewing environments, including:

- Viewing in the Procedure Editor
- Flight following in the MCC and in support environments
- Viewing in an office environment
- Viewing on hand-held devices onboard the vehicle
- Viewing printed versions of procedures.

It is vital that users can recognize a procedure in all of these varying environments as the same procedure. The consistency and uniformity enforced during procedure authoring (Section III) will enable the consistent appearance and content of procedures in multiple environments.

3. Integration with Vehicle Systems

Safety is a critical concern for procedure interactions. If procedure execution software directly controlled a vehicle using a procedure data file, then the procedure data file would be considered high criticality flight software and require months of testing before use. Since procedures are routinely updated, often during missions, this is unacceptable. To avoid this issue, the procedure execution never directly affects the vehicle. Instead, it prompts the

crew to perform the appropriate actions. The crew can choose to skip or change the action. This makes Constellation procedures fall under the same class of data as paper procedures, with the same kind of testing requirements.

a. Command Cueing

Command cueing happens when an instruction contains an explicit or implied command. The element in the associated vehicle display that the command is associated with is highlighted, and the actual command is cued. The user simply has to select the default control to send the command. But, as was discussed earlier in Section II.1.b, the user can choose to send a different command or not send a command at all (see Figure 6).

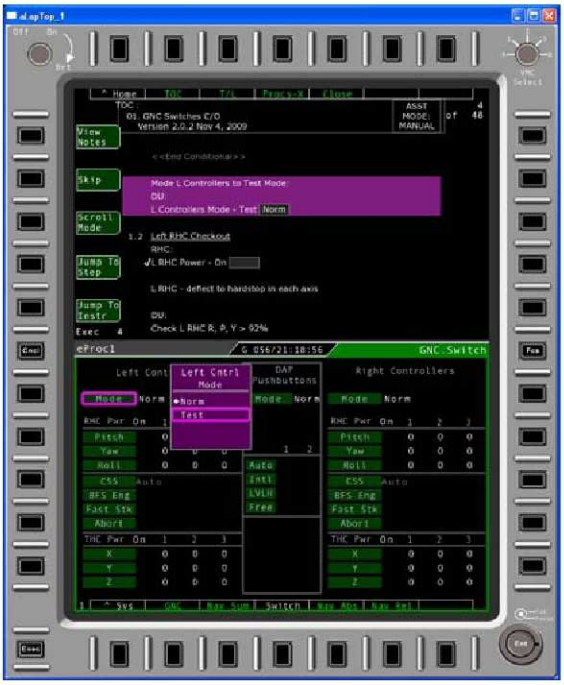

Figure 6: Proposed onboard integrated procedure display and vehicle display. *The highlighted instruction (top portion of the Display Unit) is a command to set the Left Controllers Mode to Test. The Left Controllers Mode element in the vehicle display (bottom portion of the Display Unit) is highlighted, and the command "Test" for that element is cued. If the crewmember presses the default control in the lower right corner, the command will be sent, focus will return to the procedure display, the instruction will be marked as completed, and the next instruction will be selected.*

b. Embedded Data Display

The procedure executor will display vehicle system data needed for procedure operation, e.g., to determine the state of a system or success of a command, in the procedure executor as part of the instructions. This allows the crew to perform evaluations without shifting their focus from the procedure to a system display and back. Also, this feature will be integrated with the assisted navigation feature (see Figure 4).

c. Assisted Navigation

The procedure executor will keep track of the current state of a procedure, will enable "go to" jumps to locations within the procedure, and will enable jumps to locations in other procedures. Also, the executor will monitor and evaluate success of instruction execution where possible, and advance to the next instruction somewhat automatically. A general rule is that the executor can advance automatically only if the user input was a confirmation that the instruction has been performed.

V. Procedure Verification

The CxPASS team has done detailed analyses of the verifications that will need to be performed on Constellation procedures. In general, the verification falls into three general categories, listed below with some examples of specific verifications that fall into each category:

- Internal Verifications

- Validate the procedure against the PRL schema
- Perform type checking of all values for internal consistency
- References to elements on a vehicle display must be associated with the proper vehicle display
- Path checking—there must be no unreachable instructions, not infinite loops, etc.
- Onboard executable file size—the size of an Onboard Binary Format file generated from a procedure cannot exceed the file size limits for the onboard systems

- External Inter-Procedure Verifications
  - All references to other procedures must be correct
  - All references to steps in other procedures must be correct

- External Data Verifications
  - All procedures in a given flight load must be verified against the same (correct) set of data products
  - All references to vehicle displays and display elements must be correct with regard to the flight software loads
  - All references to data values on a vehicle display must use the same type and format for that data value as the system display uses

1. Automated Verification

The Procedure Translation Verifier (PTV) application will perform syntax and logical consistency checks to verify the procedures. This tool will be used throughout the procedure lifecycle. It is recognized that there are times when verification failures will be allowed, for example when checking a partially written procedure, or when creating an incomplete flight load for a training exercise. Also, it has been determined that there will not be any automated correction of issues found during verification. That will require human re-authoring.

There will be enough information to perform all internal verifications and external data verifications using the PRL schema and the external command, telemetry, and display databases. All external inter-procedure verifications can be performed after flight loads are created.

Procedures cannot be validated with automated tools. The intent for Constellation is to have procedures validated in the same manner as static procedures are currently validated for Shuttle and ISS—through training exercises and reviews. Direct testing through training in flight-like environments will be especially important since the Constellation procedures will have a large amount of data not amenable to casual inspection, such as embedded telemetry identifiers.

The external data verification will be sufficient to determine that a procedure is capable of executing properly onboard a Constellation vehicle. Note that, just as there are multiple releases of procedures, there will be multiple releases of vehicle flight software and ground flight software. Procedures will have to verified against specific "flight loads" to ensure that they will run correctly with the software and hardware used on a specific mission. This is, however, similar to the processes used to manage static procedures for Shuttle and ISS today.

2. Improved Quality

Internal consistency of the procedures will be a hallmark of Constellation procedures. This is a large part of what will allow automated verification to be performed, and is another indication of the improved quality that will be achieved. Reviews of current procedures often uncover minor errors in procedures that have been used on missions, sometimes many missions. The human flight crew and flight controllers work around these errors as part of their daily business. The automated verification will eliminate a large number of simple errors, such as references to non-existent items or mis-numbered steps. But again, as stated above, no amount of internal consistency or automated verification can prove that a procedure will accomplish what it is supposed to accomplish, and do it without untoward consequences. The need for human validation will likely never go away in human space flight, where the demand for quality is so high.

VI. Conclusion

The approach being taken for Constellation procedures integrates crew actions that are embodied in procedures and vehicle control that is embodied in controls and displays to a degree never before seen. It is hoped that this approach will both improve the safety and quality of vehicle control, and save time and effort by the crew that can be used for other purposes. The unified and consistent procedure content captured during authoring allows automated verification as well as viewing and execution in multiple environments. The approach also provides a method that balances between human-controlled processes and automated processes through the concept of

computer actions that require human confirmation. These new concepts are grounded in decades of experience with human space flight operations. It is a large change, but one that progresses in a direction that has been accepted by a very large group of flight controllers, astronaut crewmembers, managers, and developers.

Finally, it should be noted that procedures and timelines are not planning: they are the results of planning. It is possible that in future long-term missions, automated planning tools could be coupled with the procedure execution system to manage a vehicle or habitat. For now, manned vehicle planning will remain a human process for safety reasons.

VII. Acknowledgments

The authors would like to thank the Johnson Space Center Flight Operations community, the JSC Mission Operation Directorate Procedure Management Group, the CxPASS development team of the JSC Engineering Directorate, the Exploration Technology Development Program/Automation for Operations, and the NASA Constellation Program for their work on the project that this paper discusses.

VIII. References

Software Architecture for the CxPASS System, Draft 0.104 (*official NASA document number has not yet been assigned*)
Software Requirements Specification for the CxPASS Common Subsystem, Draft 0.7 (*official NASA document number has not yet been assigned*)
Software Interface Control Document fr the CxPASS System, Draft 0.03 (JSC-61780)
Software Requirements Specification for the CxPASS/PAT Subsystem, Draft 0.27 (JSC-61756)
Software Design Document for the CxPASS/PAT Subsystem, Draft 0.61 (JSC-61781)
Software Requirements Specification for the CxPASS/PVE Subsystem, Draft 0.30 (JSC-61779)
Procedure Representation Language XML Schema, Version 3.0.03, ER6, JSC