Putting Integrated Systems Health Management Capabilities to Work: Development of an Advanced Caution and Warning System for Next-Generation Crewed Spacecraft Missions

Robert S. McCann[1] and Lilly Spirkovska[2]
NASA Ames Research Center, Mountain View, CA, 94035, USA

Irene Smith[3]
Stinger Ghaffarian Technologies Inc., Mountain View, CA, 94035, USA

Integrated System Health Management (ISHM) technologies have advanced to the point where they can provide significant automated assistance with real-time fault detection, diagnosis, guided troubleshooting, and failure consequence assessment. To exploit these capabilities in actual operational environments, however, ISHM information must be integrated into operational concepts and associated information displays in ways that enable human operators to process and understand the ISHM system information rapidly and effectively. In this paper, we explore these design issues in the context of an advanced caution and warning system (ACAWS) for next-generation crewed spacecraft missions. User interface concepts for depicting failure diagnoses, failure effects, redundancy loss, "what-if" failure analysis scenarios, and resolution of ambiguity groups are discussed and illustrated.

I. Introduction

SYSTEMS engineers rarely encounter design projects as challenging and complex as a crewed spacecraft. Particularly during the dynamic mission phases of launch, ascent, and entry, onboard propulsion systems must store, transport, mix, and detonate high volumes of extremely volatile substances under extremely harsh operational conditions. Inside the vehicle, the environmental control and life support system (ECLSS) must maintain ambient temperatures at comfortable levels for crewmembers and sensitive equipment alike, while ensuring a continuous supply of breathable air, potable water, and consumable food. Operating these onboard systems would be impossible without a command and data handling system that crunches numbers continuously from hundreds of onboard sensors, and issues flight-critical instructions to a wide variety of end effectors and automated system controllers. Last but not least, virtually all spacecraft systems interface extensively with an onboard electrical power system (EPS) that generates, stores, and distributes power to scores of fans, pumps, heaters, valves, computing devices, and many other forms of powered equipment.

The combination of high complexity, high degree of interconnectedness, and extreme operating environments makes systems malfunctions a ubiquitous risk to mission success and crew safety. For example, of the seven Apollo missions whose goal was to land crewmembers on the Moon, fully five experienced mission-threatening systems malfunctions. Reducing the risk posed by systems malfunctions influences almost all aspects of a manned spaceflight program, from the earliest stages of systems design, to real-time mission operations, to post-mission vehicle inspection and maintenance operations. Wherever possible, systems designers build in functional redundancies (backups) that provide opportunities to maintain or restore critical systems operations in the event of a failure in a primary component. Extensive failure modes, effects, and criticality analyses (FMECAs) are conducted to identify failures and trace their functional and operational consequences. Initially, these FMECAs are used to verify and validate systems designs and functional redundancies. Later, the FMECAs support the development of

[1] Research Psychologist, Human Systems Integration Division, Mail Stop 2623-4, Moffett Field, CA. AIAA Senior member.
[2] Group Lead, Intelligent Systems Division, Mail Stop 269-3, Moffett Field 94035.
[3] Software Developer, Intelligent Systems Division, Mail Stop 269-3, Moffett Field, CA. AIAA Senior Member.

step-by-step fault management procedures to minimize the impact of a malfunction and, where possible, recover lost functionality by exploiting redundancies.

Once the spacecraft is built and launched, monitoring, managing, and maintaining the health of the onboard systems assumes a prominent role in real-time mission operations. Each onboard system is richly instrumented with sensors that provide real-time numeric readings of critical operating parameters, such as temperatures, pressures, voltages, and flow rates. If a sensor starts returning values outside of upper and lower limits that define the range deemed consistent with nominal operation, the vehicle's Caution and Warning (C&W) system automatically responds with multiple auditory and visual indications. If, as is often the case, the failure propagates to other components or systems, those entities add their own indications. The result is a cascade of C&W system alarms, multiple off-nominal indications on cockpit and ground-based system summary displays, and lengthy lists of fault messages on C&W system fault logs. Before the crew can start executing appropriate fault isolation and recovery procedures, flight controllers must act as diagnostic agents, processing, cross-checking, and evaluating the C&W system indications to identify the "parent" malfunction, and understand what downstream entities ("children") have either stopped operating completely, or are operating in an unintended and possibly dangerous manner. An example of the latter would be if an electrical switch "failed closed", thereby completing a circuit that powers (opens) an electrically actuated valve that then allows hypergolic fuels to mix in the combustion chamber of a spacecraft thruster. The immediate result would be an unplanned change to spacecraft attitude and velocity that, if the vehicle was engaged in a rendezvous and docking operation, could quickly become a mission-threatening situation. In this circumstance, the most appropriate immediate response might be to manually close the valves that control the flow of fuel to the thruster from the storage tanks, as that action would shut down the thruster immediately, rather than troubleshooting and dealing with the parent itself (the failed switch).

Diagnosing the source of real-time failures, and then deciding upon the appropriate set of isolation and recovery activities, are only one facet of a controller's health-management responsibilities. Flight controllers who sit on console in the main flight control room and report to the flight director must maintain a high degree of situation awareness of vehicle state and system status at all times, both to anticipate off-nominal situations that may develop and to maximize the efficiency and accuracy of troubleshooting and decision-making should a real problem arise. After fault isolation and recovery procedures have been completed, flight controllers continue to engage in extensive analyses to understand what flight-critical systems components have lost redundancy and are now "zero fault tolerant".

In today's operational environment, maintaining the high levels of situation awareness needed to meet these ongoing health-management-related requirements can be difficult. Along with the real-time information about systems health and status that is depicted on electronic displays of spacecraft telemetry, pertinent information is often distributed across a variety of paper engineering products, such as cue cards, systems schematics, flight rules, and procedures. Many of these products have been developed in a "stovepipe" fashion for a specific training purpose or by a specific engineering division at NASA's Johnson Space Center, and are therefore customized to the point where flight controllers must exert considerable mental effort to integrate the information contained in one product with the information in the others. Sometimes, some of these products are only available to the systems experts who sit in mission support "backrooms" such as the Mission Evaluation Room (MER), meaning the flight controllers have no direct access to the products themselves. The net result is that the current set of engineering products make it difficult to achieve and maintain a high degree of shared situation awareness between flight controllers, flight directors, and backroom staff members.

After 50 years of manned spaceflight operations confined to the Earth-Moon system, NASA is finally building the infrastructure to support crewed missions to much more remote destinations such as Mars. These missions will require fundamental changes to mission operations for both crew and ground personnel. Onboard, speed-of-light limitations will force crewmembers to diagnose and work the most time-critical systems malfunctions with either highly degraded or nonexistent ground support. Current-generation (i.e., Space Shuttle-era) caution and warning systems, cockpit systems displays, and onboard automation are woefully inadequate to provide the crew with these capabilities. As for the ground, a recent evaluation of the effect of time delay on mission operations in the DSH[1] revealed that trying to conduct mission operations under significant time delay with today's set of mission support products led to ratings of workload by flight controllers that often fell in the "unacceptable without improvement" range of the Bedford subjective workload rating scale. Moreover, given the extended duration of these missions, rather than spending dedicated periods of time in the MERs, many backroom systems experts will shift to "on-call" availability, which means that when a flight controller requests their services, they will need work products that get them "up-to-speed" on current vehicle health and operational status as quickly and efficiently as possible. Today's collection of decision-support products are not well suited to support these next-generation missions.

Fortunately, over the past two decades, applied artificial intelligence technologies associated with the emerging field of integrated systems health management (ISHM)[2] have matured to the point where they can support a more integrated, and more capable, set of decision support products for crewmembers and flight controllers alike[3]. One of the core technologies of the ISHM discipline is a formal systems model that captures functional relationships and functional dependencies between system elements[4-7]. Such models can support a wide variety of decision support tools; systems analyses; failure modes, effects, and criticality analyses; development and validation of fault management procedure and flight rules; automated failure detection, isolation, and recovery systems; training; and operations support tools. For example, the Testability, Engineering, and Maintenance System (TEAMS) tool developed by Qualtech Systems, Inc.[8] assesses the health of a system on a continual basis, and automatically tries to diagnose the source (or "parent") failure of a C&W event. TEAMS associates each component in a system with "test points", sensor readings whose values depend directly on the operating mode of the component in question. Consider a hypothetical and very stripped-down example of an ECLSS pump whose power comes through a Remote Power Controller (RPC) in a spacecraft's EPS. Suppose the EPS is instrumented with a sensor that measures the current on the bus connecting the RPC to the pump. If the RPC were to fail off, the current sensor would start returning an out-of-limits low value (assuming the sensor itself was healthy and operators were receiving live telemetry). If the RPC was functioning as intended, and distributing power to the pump, the current sensor would be returning values within normal limits. Because of these straightforward dependencies between the behavior of the RPC and the behavior of the current sensor, TEAMS modelers would include the current sensor as a "test point" for determining the health of the RPC.

Suppose now that the pump itself has two test points: A sensor that measures pump RPM, and a "downstream" sensor that measures the flow rate of the fluid being driven by the pump. All of a sudden, some or all of these sensed values go out of limits, triggering caution and warning alerts and error messages through the conventional caution and warning system. The ground will want to determine as soon as possible whether this C&W "event" is caused by an RPC failure, a pump failure, one or more sensor failures, or some combination of the above. If all three test points are returning off-nominal readings (i.e. current is showing out-of-limits low, the Pump RPM is showing out-of-limits low, and the flow rate downstream of the pump is showing out-of-limits low), TEAMS will diagnose the RPC as the minimal component that best explains all the failure indications. Alternatively, if the current sensor is returning a normal value, but both the RPM and the flow rate are showing out-of-limits low, the nominal status of the value being reported by the current sensor will exonerate the RPC as a candidate to explain test point behavior, and the pump will be declared failed instead. Finally, if just one of the two pump test points –say, the RPM value – has gone out-of-limits, TEAMS will judge the RPM sensor as failed.

Scaled up from this simple example, a fully developed TEAMS model can form the basis of an ACAWS with the following capabilities:

- Identify system effects of a failure, where system effects correspond to the set of downstream components whose operational status has departed from nominal in some way due to a functional connection with the parent failure. The most straightforward form of system effect is a downstream component that simply stops operating, such as what would happen to our ECLSS fan if the RPC responsible for routing power to the fan failed OFF.
- Determine the impact of a failure on scheduled timeline activities for which the onboard systems and their capabilities are essential resources.
- Identify cases where the existing test points are inadequate to unambiguously identify a failure given its failure signature. In such situations, TEAMS outputs the set of components (called an ambiguity group) whose failures are all consistent with the current pattern of test results.
- Point the end user to the correct set of manual tests (if any are available) that will provide the model with sufficient additional operational information (additional test points) to disambiguate the problem and resolve the ambiguity group.
- Allow users to inject their own failures in "what-if" exercises and assess these failures' impacts on the health of other components (e.g., "if I fail element X of a system, what downstream elements are affected as a result?").
- Determine, in cases where the failure signature results in a diagnostic ambiguity group, the "common" set of downstream components that no longer function regardless of how the ambiguity group may be resolved (i.e., regardless of which member of the ambiguity group is determined, through additional testing, to be the failure).
- Determine, for an ambiguity group, the set of downstream effects that are "possible" impacts, that is, may or may not remain impacted depending on the outcome of the ambiguity group resolution. System

components that fall in the "possible" category have functional dependencies with some, but not all, of the suspects in the ambiguity group.

- Determine, for targeted failures, what elements of the system have now lost functional redundancy and are at increased risk of being impacted by another upstream failure. Determining loss of redundancy is useful for identifying components that are at heightened vulnerability and as a first step to identifying and preparing for the "next worst failure" that could occur.

As we've noted, in today operational environment, accessing the information to make these determinations and to train and enhance situation awareness with explorations of "what-if" failures and their consequences often requires accessing and mentally integrating information from a wide variety of engineering products, some electronic (sensor data) and others paper-based. A golden opportunity exists to consolidate the current set of information displays and paper products into a single integrated decision support and information display system. Of course, developing such a system presents several challenges in the area of user interface design and information display. In the rest of this article, we illustrate these challenges, and describe a candidate set of display and user interface designs, for an Advanced Caution and Warning System (ACAWS).

II. ACAWS

A. ACAWS Overview.

ACAWS is a comprehensive system health management tool composed of software modules that work in tandem to help spacecraft operators monitor the health of a system, detect anomalous health conditions, diagnose the cause of off-nominal detected conditions, and link to procedures for recovering from a failure or mitigating its effects to accomplish as much of the planned mission as possible.

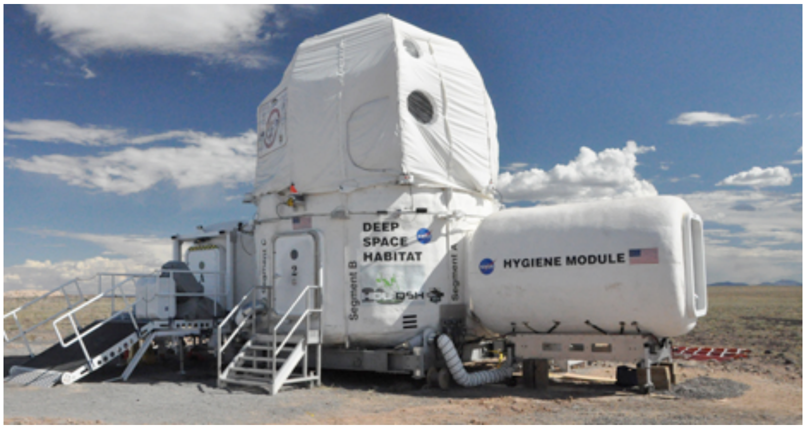


Figure 1. Image of the Deep Space Habitat deployed in the Arizona Desert. The main Lab module has "Deep Space Habitat" painted on its surface.

For development purposes, we have focused our efforts on modeling the EPS of NASA Johnson Space Center's Deep Space Habitat (DSH), an earth-analog of a workspace and living area that might house a crew during the transport and surface phases of a deep-space crewed mission[9]. Shown in Fig. 1, the DSH is comprised of several modules. The main "Lab" module (labeled "Deep Space Habitat" in the figure), is divided into eight pie-piece shaped work areas including a Medical Operations Work Station (MOWS), a Tele-Robotics Work Station (TRWS) and a General Maintenance Work Station (GMWS). The primary power used by the DSH is 120 VAC supplied from a generator. Power is distributed to the various loads through a set of Power Distribution Units (PDUs), each housing 16 ports arranged in two banks of eight ports each. Secondary power sources of 120 VAC, 28 VDC, and 120 VDC are also available for use. Power receptacles at 120 VAC are located both internal and external to the DSH for powered equipment, such as a vacuum cleaner or power tools. Instrumentation system sensors are located in each of the DSH modules and connecting airlock subsystems, thus providing test points for the ACAWS diagnostic reasoner. These sensors are powered by the DSH power.

One of the PDUs, PDU-B1, is exceptional for having functional connections to a particularly wide variety of components. Shown in a hierarchical arrangement in Fig. 2, some components, like the two power converters (a 24VDC and a 28VDC), are part of the EPS system itself. Others, such as the compact Reconfigurable Input/Outbox (cRIO) box, are associated with the Avionics system. Still others, such as the Wireless Sensor Nodes (WSNs) are part of the Command and Data Handling system. The remaining components are primarily loads (e.g., the powered drills, saws, etc. associated with the GMWS, and the solid state light modules [SSLMs] that provide ambient illumination in the Lab Module).

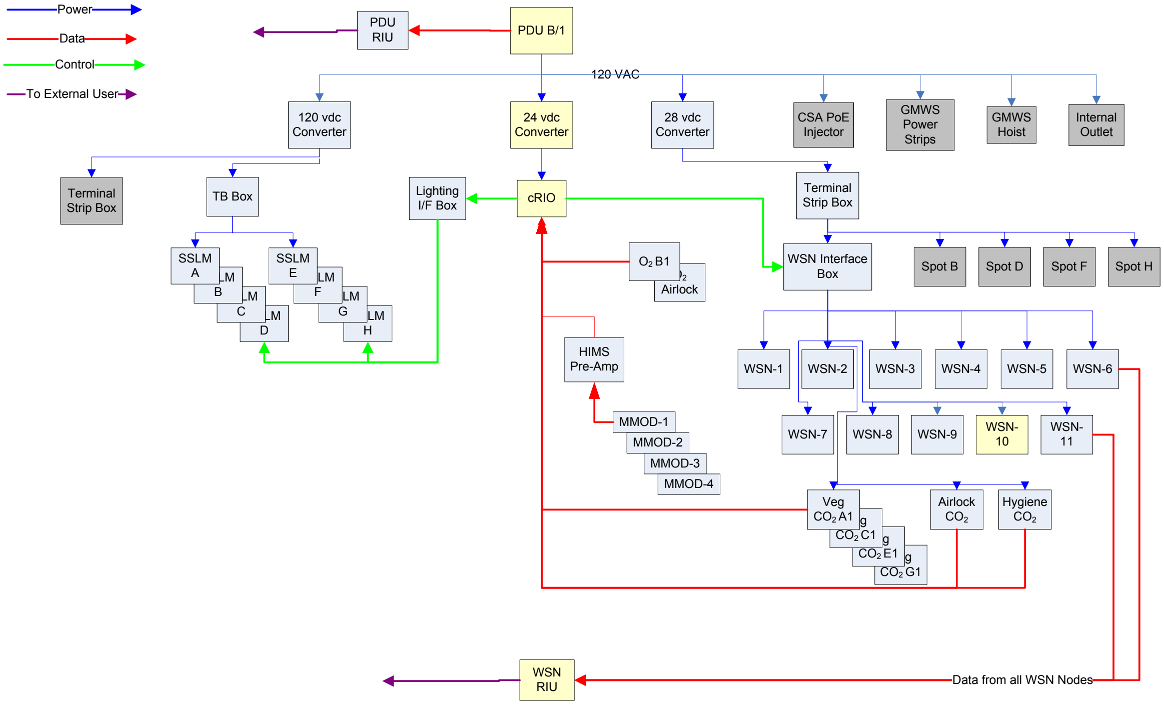


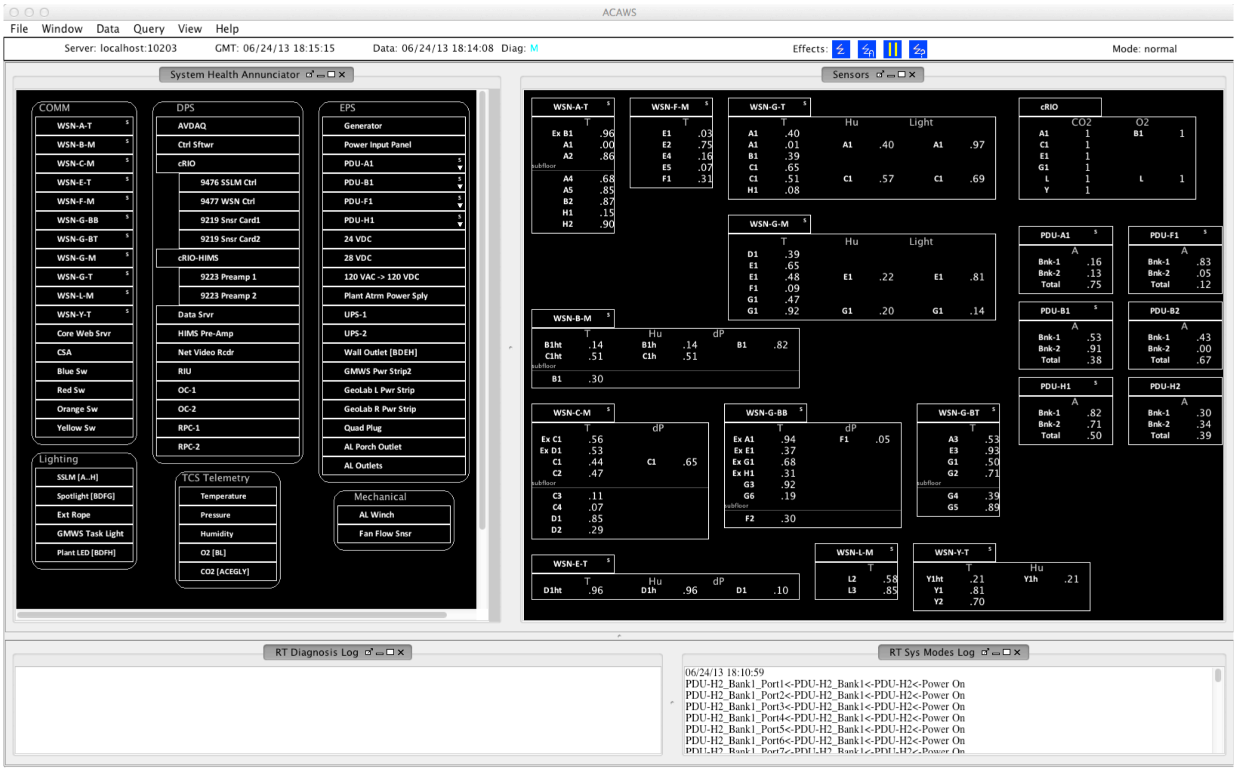
Figure 2. PDU B1 and dependent elements of the DSH.

ACAWS determines the diagnostic state of these components using Qualtech Systems, Inc.'s TEAMS-RDS runtime diagnostic engine that relies upon test points compiled from a user-generated, domain-specific TEAMS model of the DSH EPS. In addition to the TEAMS diagnostic reasoning engine, ACAWS has a SysEffects module that computes how failures propagate and manifest through the system. A number of component failures associated with the PDU-B1 power string have very similar patterns of test results, complicating the determination and annunciation of the failure. For instance, when presented with C&W messages associated with a failure along the PDU-B1 power string, it is not immediately obvious whether the C&W event originates in PDU-B1 itself, in the two pairs of redundant ports that provide power to the 24 VDC and 28 VDC converters, respectively, in the 24 VDC or 28 VDC converters themselves, in the cRIO, the WSNs, or a Remote Interface Unit (RIU; software that reads and publishes WSN data).

The rather large "problem space" made the PDU-B1 string an attractive target for exercising and illustrating ACAWS capabilities. We now illustrate these capabilities, and the user interface features that make them available to the operator, by invoking a hypothetical novice DSH flight controller who first employs ACAWS in a "what-if" mode to explore and become familiar with PDU-B1-related failure modes and their impacts. Later, when the novice has been trained sufficiently to sit on console as a flight controller, we illustrate how she might use ACAWS for real-time failure diagnosis, isolation, and recovery activity.

ACAWS provides a window pane layout framework that facilitates display customization. An operator can arrange the display to best support the current task. For instance, a "monitoring" display layout provides an overview of system health when operations are nominal, and an "analysis" display layout that provides panes to "drill down" to more detailed information sources when analyzing failures and failure effects. An example of a possible configuration of the ACAWS displays designed for monitoring mode is shown in Fig. 3. The left panel, a general systems health annunciator panel, groups top-level systems elements into bundles at the system level, and depicts key high-level elements of those systems as rectangular objects nick-named "chiclets". In some cases, subelements of a "higher-level" element are depicted on the display as indented chiclets below the "parent"; this is the case for the four data-handling cards contained in the cRIO, for example. In other cases, such as the PDUs in the EPS grouping, the numbers and downpointing arrows on the right of the PDU chiclets indicate that an expanded set of chiclets are available for viewing underneath the PDUs themselves via a popup menu of display options.

The right pane will be familiar to controllers who like to scan displays of raw vehicle telemetry to maintain situation awareness of system functioning down at the level of individual sensor values. The bottom left pane is


ACAWS
File
Window
Data
Query
View
Help
Effects:
System Health Annunciator
COMM
WSN-A-T
WSN-B-M
WSN-C-M
WSN-E-T
WSN-F-M
WSN-G-BB
WSN-G-BT
WSN-G-M
WSN-G-T
WSN-L-M
WSN-Y-T
Core Web Srvr
CSA
Blue Sw
Red Sw
Orange Sw
Yellow Sw
DPS
AVDAQ
Ctrl Sftwr
cRIO
9476 SSLM Ctrl
9477 WSN Ctrl
9219 Snsr Card1
9219 Snsr Card2
cRIO-HIMS
9223 Preamp 1
9223 Preamp 2
Data Srvr
HIMS Pre-Amp
Net Video Rcdr
RIU
OC-1
OC-2
RPC-1
RPC-2
EPS
Generator
Power Input Panel
PDU-A1
PDU-B1
PDU-F1
PDU-H1
24 VDC
28 VDC
120 VAC -> 120 VDC
Plant Atrm Power Sply
UPS-1
UPS-2
Wall Outlet [BDEH]
GMWS Pwr Strip2
GeoLab L Pwr Strip
GeoLab R Pwr Strip
Quad Plug
AL Porch Outlet
AL Outlets
Lighting
SSLM [A..H]
Spotlight [BDFG]
Ext Rope
GMWS Task Light
Plant LED [BDFH]
TCS Telemetry
Temperature
Pressure
Humidity
O2 [BL]
CO2 [ACEGLY]
Mechanical
AL Winch
Fan Flow Snsr
Sensors
WSN-A-T T Ex B1 .96 A1 .00 A2 .86 subfloor A4 .68 A5 .85 B2 .87 H1 .15 H2 .90
WSN-F-M T E1 .03 E2 .75 E4 .16 E5 .07 F1 .31
WSN-G-T T A1 .40 A1 .01 B1 .39 C1 .65 C1 .51 H1 .08 Hu A1 .40 C1 .57 Light A1 .97 C1 .69
cRIO CO2 A1 1 C1 1 E1 1 G1 1 L 1 Y 1 O2 B1 1 L 1
WSN-G-M T D1 .39 E1 .65 E1 .48 F1 .09 G1 .47 G1 .92 Hu E1 .22 G1 .20 Light E1 .81 G1 .14
PDU-A1 A Bnk-1 .16 Bnk-2 .13 Total .75
PDU-F1 A Bnk-1 .83 Bnk-2 .05 Total .12
WSN-B-M T B1ht .14 C1ht .51 subfloor B1 .30 Hu B1h .14 C1h .51 dP B1 .82
PDU-B1 A Bnk-1 .53 Bnk-2 .91 Total .38
PDU-B2 A Bnk-1 .43 Bnk-2 .00 Total .67
WSN-C-M T Ex C1 .56 Ex D1 .53 C1 .44 C2 .47 subfloor C3 .11 C4 .07 D1 .85 D2 .29 dP C1 .65
WSN-G-BB T Ex A1 .94 Ex E1 .37 Ex G1 .68 Ex H1 .31 G3 .92 G6 .19 subfloor F2 .30 dP F1 .05
WSN-G-BT T A3 .53 E3 .93 G1 .50 G2 .71 subfloor G4 .39 G5 .89
PDU-H1 A Bnk-1 .82 Bnk-2 .71 Total .50
PDU-H2 A Bnk-1 .30 Bnk-2 .34 Total .39
WSN-E-T T D1ht .96 Hu D1h .96 dP D1 .10
WSN-L-M T L2 .58 L3 .85
WSN-Y-T T Y1ht .21 Y1 .81 Y2 .70 Hu Y1h .21
RT Diagnosis Log
RT Sys Modes Log

Server: localhost:10203 GMT: 06/24/13 18:15:15 Data: 06/24/13 18:14:08 Diag: M Mode: normal

```
06/24/13 18:10:59
PDU-H2_Bank1_Port1<-PDU-H2_Bank1<-PDU-H2<-Power On
PDU-H2_Bank1_Port2<-PDU-H2_Bank1<-PDU-H2<-Power On
PDU-H2_Bank1_Port3<-PDU-H2_Bank1<-PDU-H2<-Power On
PDU-H2_Bank1_Port4<-PDU-H2_Bank1<-PDU-H2<-Power On
PDU-H2_Bank1_Port5<-PDU-H2_Bank1<-PDU-H2<-Power On
PDU-H2_Bank1_Port6<-PDU-H2_Bank1<-PDU-H2<-Power On
PDU-H2_Bank1_Port7<-PDU-H2_Bank1<-PDU-H2<-Power On
```

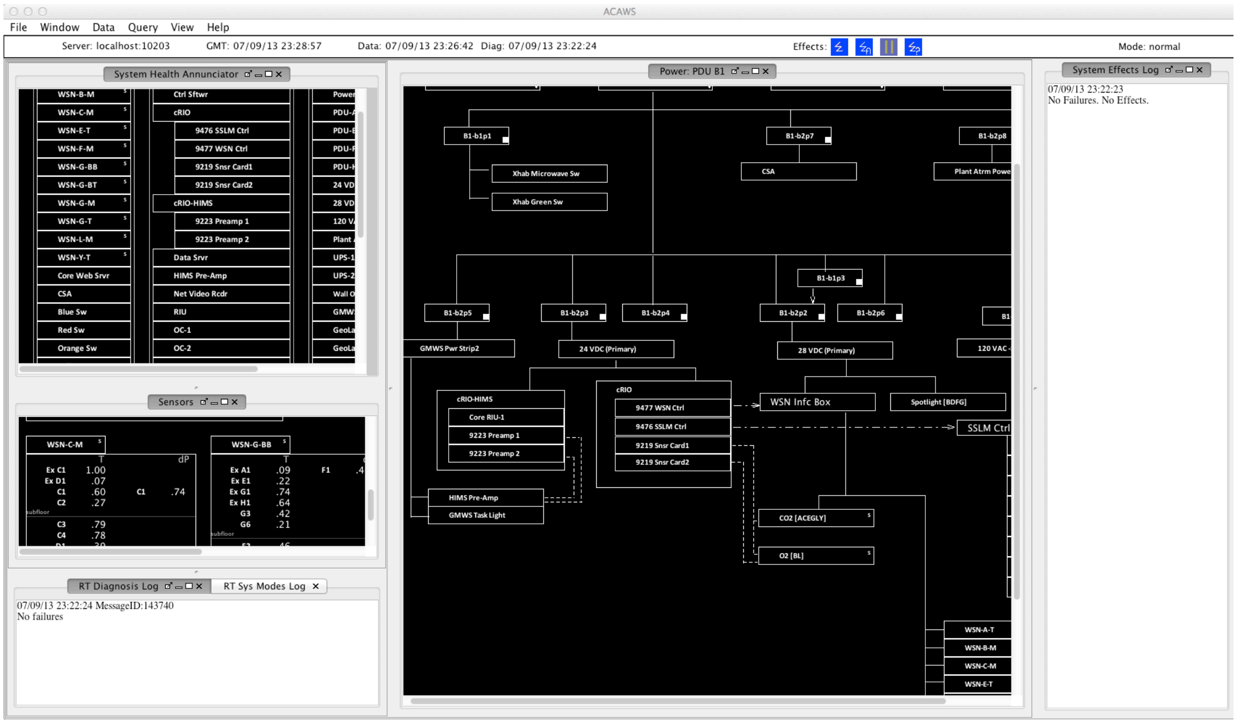
Figure 3. Candidate selection and arrangement of ACAWS panes for "Monitor" mode. The Systems Health Annunciator on the left groups system components and depicts them as "chiclets" (see text for details). Selecting a chiclet brings up a pop-up menu of ACAWS choices. The right pane depicts sensor values. The bottom Left pane is blank, as no off-nominal entries exist in the RT Diagnosis Log. The bottom Right pane contains a log of currently commanded port modes.

blank, as the system is functioning is nominal mode, and the bottom right pane provides an indication of the status of the ports within each PDU (Commanded "On" or Commanded "Off").

B. ACAWS Capabilities and Illustrative Cases

The set of icons colored "blue" on the ACAWS display status bar (top of the display) provide ways for the user to exploit the various capabilities offered by ACAWS. Suppose our novice flight controller wanted to use ACAWS to learn more about failure modes associated with PDU-B1 and associated DSH components. The first thing she might do is move and minimize the size of the sensor display pane, replacing it with the more analytic (and PDU-B1-centric) Power: PDU-B1 display shown in Fig. 4. This display depicts selected EPS and other systems components in a hierarchical arrangement that allows the user to trace dependencies from the highest levels of the power system all the way to the loads (consumers) of the power. Three forms of dependencies between components are depicted: Power dependencies are illustrated through continuous lines, data-sharing dependencies through dashed lines, and unidirectional commanding dependencies through broken "dot-and-dash" lines terminating in arrows (showing the direction of the commanding).

Suppose our novice first decides to explore the impacts and implications of failing the 28 VDC power converter. To begin exploration of this situation, she first "right clicks" on the 28 VDC chiclet in either the "Systems Health Annunciator" pane or the "Power: PDU B1" pane. Right clicking brings up a popup menu (not shown in the Figure) with a variety of selectable options, one of which is "Suppose Health: Failed". Selecting "Suppose Health: Failed" automatically switches the display from "Mode: normal" (Fig. 4) to "Mode: hybrid" (Fig. 5), which supports display of failures and effects that have been either "user-failed" or "naturally failed" (where "naturally" refers to failures annunciated by ACAWS on the basis of analyzing actual telemetry from an actual mission). "Mode: hybrid" has a number of visual features to distinguish it from "Mode: normal". The status bar and the System effects log pane are colored orange, and all components marked with the ACAWS blue square (see below) contain a small orange "F" (for "User-Failed"). Note that selecting "Suppose Health: Failed" for a chiclet is only one of several user actions that



Server: localhost:10203 GMT: 07/09/13 23:28:57 Data: 07/09/13 23:26:42 Diag: 07/09/13 23:22:24 Mode: normal

07/09/13 23:22:24 MessageID:143740
No failures

07/09/13 23:22:23
No Failures. No Effects.

Figure 4. Candidate ACAWS pane selection and arrangement for "analytic" mode. Power: PDU-B1 display has been brought up on the center-right pane. Power: PDU-B1 depicts systems components in a hierarchical arrangement that allows the user to trace dependencies from the highest levels of the power system all the way to the loads (consumers) of the power. See text for more details.

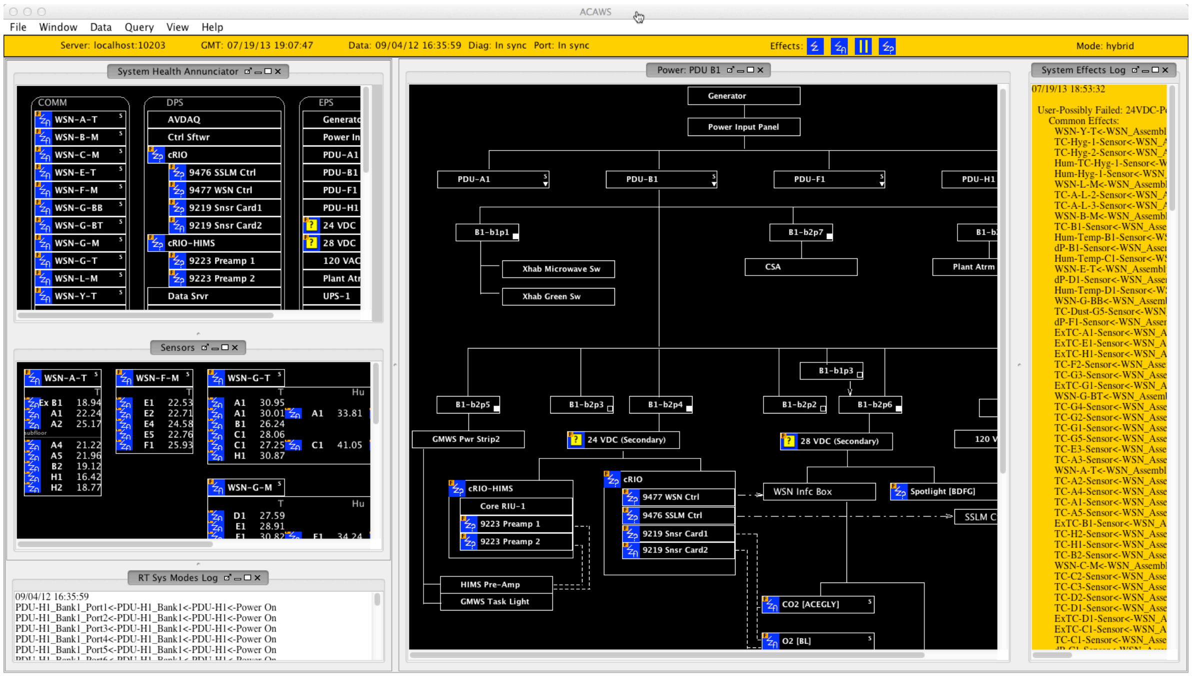
will convert a display from "Mode: normal" to "Mode: hybrid"; for example, users also have the option of overriding a port setting by right-clicking on a "port" chiclet, selecting "Suppose Switch Mode" and then selecting ON or OFF from the pop-up menu.

In Fig. 5, our novice has "user-failed" the 28 VDC. As a result, a large number of components, on the Power: PDU B1 display, the Sensors display, and the System Health Annunciator display, now host embedded blue rectangles. The presence of these rectangles, and their color, indicate that the components have been "tagged" by ACAWS (in general, the color blue is a reserved code for ACAWS-derived information). Each blue rectangle has embedded symbology. Both the 28VDC and the 24VDC chiclets contain a smaller filled yellow rectangle with a black question mark ("?"). The presence of multiple question marks indicates to our novice that ACAWS has determined that a failure of either component is consistent with (or, in other words, explains equally well) the current pattern of test results. The two power converters thus form an ambiguity group.

Downstream of the two elements in the ambiguity group, all embedded ACAWS rectangles contain a rotated and stylized letter "E", which airplane pilots sometimes use to indicate components on their plane that are not working. The stylized "E", for "effects" or "impacts", identifies downstream elements that, due to their functional connection with an upstream element, are no longer operating as designed or as desired. Earlier, we noted the most common form of impact, which is for the downstream component to have simply stopped operating. However, other forms of impact are also possible. Suppose one of the loads powered by PDU-B1 is a drill in the GMWS. If the upstream failure is a port that has stuck-ON, the drill will be powered when it shouldn't be. Or, consider the SSLMs, the set of solid-state ceiling lights that provide ambient illumination to the Lab Module. As shown on the Power: PDU B1 pane, the SSLM's are connected by command line to a control card inside the cRIO. If that control card were to fail, the lights could no longer be automatically commanded off or on by a DSH computer. They would still be fully functional, but could only be operated by a crewmember manually toggling their on/off switches.

C. ACAWS Capability Codes.

Some of the stylized "E"s are accompanied by the subscript "?" (for "may be impacted"), indicating that this effect is not written in stone; the functional pathway for the component in question traces up to one or more, but not



Server: localhost:10203 GMT: 07/19/13 19:07:47 Data: 09/04/12 16:35:59 Diag: In sync Port: In sync

Mode: hybrid

```
07/19/13 18:53:32

 User-Possibly Failed: 24VDC-P
   Common Effects:
     WSN-Y-T<-WSN_Assembl
     TC-Hyg-1-Sensor<-WSN_A
     TC-Hyg-2-Sensor<-WSN_A
     Hum-TC-Hyg-1-Sensor<-W
     Hum-Hyg-1-Sensor<-WSN_
     WSN-L-M<-WSN_Assembl
     TC-A-L-2-Sensor<-WSN_A
     TC-A-L-3-Sensor<-WSN_A
     WSN-B-M<-WSN_Assembl
     TC-B1-Sensor<-WSN_Asse
     Hum-Temp-B1-Sensor<-WS
     dP-B1-Sensor<-WSN_Asse
     Hum-Temp-C1-Sensor<-WS
     WSN-E-T<-WSN_Assembl
     dP-D1-Sensor<-WSN_Asse
     Hum-Temp-D1-Sensor<-WS
     WSN-G-BB<-WSN_Assem
     TC-Dust-G5-Sensor<-WSN
     dP-F1-Sensor<-WSN_Asse
     ExTC-A1-Sensor<-WSN_A
     ExTC-E1-Sensor<-WSN_A
     ExTC-H1-Sensor<-WSN_A
     TC-F2-Sensor<-WSN_Asse
     TC-G3-Sensor<-WSN_Asse
     ExTC-G1-Sensor<-WSN_A
     WSN-G-BT<-WSN_Assem
     TC-G4-Sensor<-WSN_Asse
     TC-G2-Sensor<-WSN_Asse
     TC-G1-Sensor<-WSN_Asse
     TC-G5-Sensor<-WSN_Asse
     TC-E3-Sensor<-WSN_Asse
     TC-A3-Sensor<-WSN_Asse
     WSN-A-T<-WSN_Assembl
     TC-A2-Sensor<-WSN_Asse
     TC-A4-Sensor<-WSN_Asse
     TC-A1-Sensor<-WSN_Asse
     TC-A5-Sensor<-WSN_Asse
     ExTC-B1-Sensor<-WSN_A
     TC-H2-Sensor<-WSN_Asse
     TC-H1-Sensor<-WSN_Asse
     TC-B2-Sensor<-WSN_Asse
     WSN-C-M<-WSN_Assembl
     TC-C2-Sensor<-WSN_Asse
     TC-C3-Sensor<-WSN_Asse
     TC-D2-Sensor<-WSN_Asse
     TC-D1-Sensor<-WSN_Asse
     ExTC-D1-Sensor<-WSN_A
     ExTC-C1-Sensor<-WSN_A
     TC-C1-Sensor<-WSN_Asse
```

```
09/04/12 16:35:59
PDU-H1_Bank1_Port1<-PDU-H1_Bank1<-PDU-H1<-Power On
PDU-H1_Bank1_Port2<-PDU-H1_Bank1<-PDU-H1<-Power On
PDU-H1_Bank1_Port3<-PDU-H1_Bank1<-PDU-H1<-Power On
PDU-H1_Bank1_Port4<-PDU-H1_Bank1<-PDU-H1<-Power On
PDU-H1_Bank1_Port5<-PDU-H1_Bank1<-PDU-H1<-Power On
```

Figure 5. Mode: hybrid ACAWS Display. Display has transitioned from Mode: normal by user clicking on "28 VDC" chiclet in either System Health Annunciator or Power PDU B1 panes and selecting "Suppose Health: Failed" option from the pop-up menu (not shown). Mode: hybrid is depicted through orange fill for the status bar and the System Effects Log and accompanying all ACAWS-tagged chiclets with a small orange rectangle containing black "F".

all, suspects in the ambiguity group. If ACAWS were to be provided enough additional test information (perhaps by manual tests; see below) to disambiguate the group, depending on which member of the ambiguity group was declared the "parent", and which member(s) was [were] exonerated, these effects might disappear. Other components declared as "effects" contain the subscripted mathematical symbol for the intersection set (the "∩" symbol). The "∩" indicates that the component is an element of the "common" set, with functional connections to all candidates in the ambiguity group. Elements in the common set will stay failed even if the ambiguity group is resolved to just one parent.

The three ACAWS codes (stylized "Ɛ", "?", and "∩") are also represented in the row of icons within the status bar along the top of the ACAWS display, along with a fourth symbol consisting of the blue rectangle with two vertical yellow stripes. The blue color of the rectangles containing these symbols indicates that the user has actively selected all four ACAWS codes to be depicted on the display if applicable, which is why three of the four codings are present among the impacts. At the user's discretion, each of the four codes can be selected or deselected (when selected, the rectangle containing the symbol is filled blue; when deselected, the rectangle is filled gray). When an impact code is deselected, the corresponding coding on the displays is removed, uncluttering the displays and allowing the user to focus on a particular class (or classes) of information (such as, for example, only the effects belonging to the common set).

The fourth symbol, a pair of vertical stripes, codes for Loss Of Redundancy (LOR) status. LOR status is computed by assigning each component in the display a value R, corresponding to the "fault tolerance" of the component: The number of distinct backup "pathways" (reconfigurations) that are available to restore the component to a functional state, should an upstream component along the currently active pathway experience a failure. Normally, there is only one redundant configuration available, so R = 1, but in some cases R > 1. Except where R = 0, meaning the component is "zero fault tolerant" to begin with, an LOR icon is displayed inside the component's chiclet whenever R ➔ R-1 due to a failure of a component in the currently active pathway. Note that when R = 0, a failure along the backup pathway will again fail the targeted component with no recourse to restore nominal function, which is why the component is now zero fault tolerant.

In Fig. 4, the "fill" color for the rectangle containing the LOR icon is gray, meaning LOR coding has been de-selected. Now suppose over time our novice controller gains considerable experience and confidence, and decides to

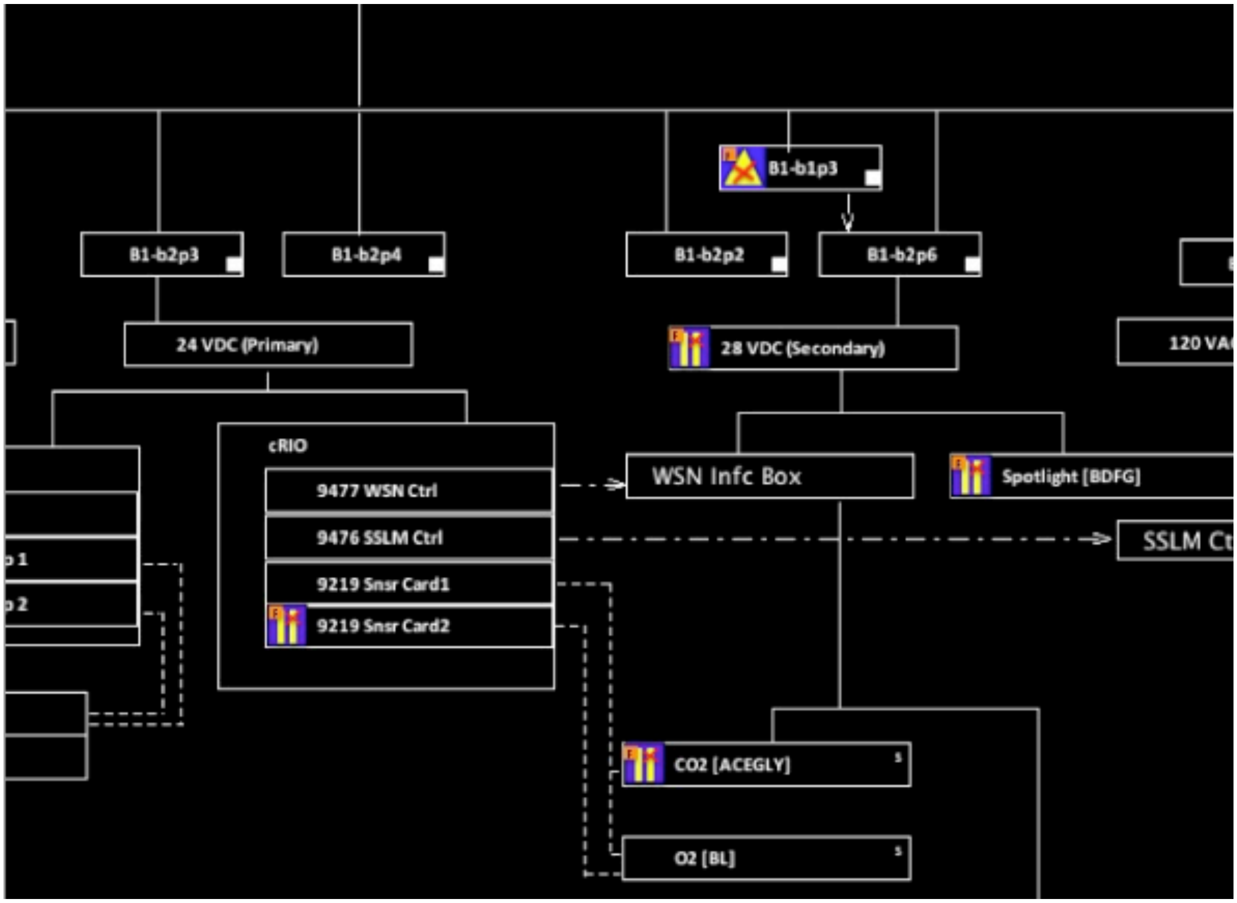
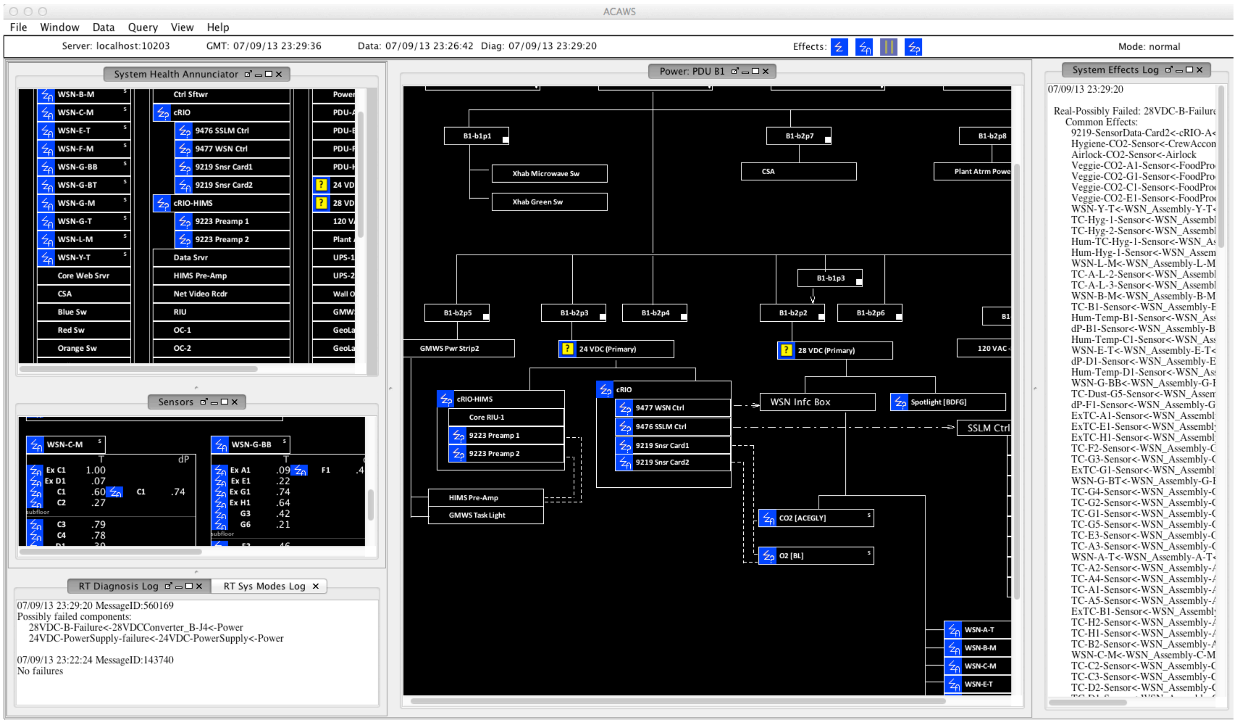


Figure 6. ACAWS Display following user deselecting ACAWS "Effects", "Common Set", and "May Be Impacted" codes and selecting "LOR" codes on Status Bar (not shown in Figure), and user failing Port B1-b1p3 OFF. The yellow triangle with overlaid red "X" in the Port B1-b1p3 chiclet is the ACAWS code for a component that's declared failed. The four components that have lost redundancy (and are now zero fault tolerant) contain the ACAWS LOR icon (two vertical stripes with a red "X" overlaid on the right-most stripe).

explore the LOR impacts of failing Port B1-b1p3 (read as PDU-B1, bank 1, port 3) to "StuckOFF". The resulting ACAWS display is shown in Fig. 6. PDU B1-b1p3 is a power control port; when commanded ON (a commanded state represented in the port chiclet by the filled white rectangle), power can only be channeled to the 28 VDC converter through primary Port B1-b2p2. If Port B1-b1p3 was commanded OFF (which would be coded in the display by an empty rather than a filled rectangle in the chiclet), power could only be channeled through the backup Port B1-b2p6.

In the configuration illustrated in Fig. 6, Port B1-b1p3 is commanded ON, but user-failed OFF. The test signature for this failure is unambiguous, so ACAWS diagnoses the port as "FailedOFF". The port chiclet thus contains the ACAWS symbol for "declared failed", a filled yellow triangle with an embedded red "X". The "FailedOFF" configuration has two functional consequences. First, consistent with the OFF mode, power now flows to the 28 VDC converter through the backup Port B1-b2p6. Second, since the "Failed OFF" status of the port means that it cannot be turned "ON", the 28 VDC can ONLY be powered through Port B1-b2p6; there is no way to reconfigure the system to get power through the primary port. Hence, if B1-b2p6 were to *also* fail, the 28 VDC and all downstream units would fail with no possibility of recovery. The components that have suffered this LOR (and are also now zero fault tolerant) are depicted with the LOR "dual stripe" symbol, with a red X superimposed on the right-hand stripe.

The insight provided by these various ACAWS codes in the "what-if" mode can assist controllers with a variety of troubleshooting situations. Knowledge of what elements belong to the common set for selected failures, for



```
Server: localhost:10203    GMT: 07/09/13 23:29:36    Data: 07/09/13 23:26:42  Diag: 07/09/13 23:29:20    Mode: normal
```

RT Diagnosis Log

```
07/09/13 23:29:20 MessageID:560169
Possibly failed components:
    28VDC-B-Failure<-28VDCConverter_B-J4<-Power
    24VDC-PowerSupply-failure<-24VDC-PowerSupply<-Power

07/09/13 23:22:24 MessageID:143740
No failures
```

System Effects Log

```
07/09/13 23:29:20

  Real-Possibly Failed: 28VDC-B-Failure
     Common Effects:
       9219-SensorData-Card2<-cRIO-A<
       Hygiene-CO2-Sensor<-CrewAccon
       Airlock-CO2-Sensor<-Airlock
       Veggie-CO2-A1-Sensor<-FoodPro
       Veggie-CO2-G1-Sensor<-FoodPro
       Veggie-CO2-C1-Sensor<-FoodProd
       Veggie-CO2-E1-Sensor<-FoodProd
       WSN-Y-T<-WSN_Assembly-Y-T<
       TC-Hyg-1-Sensor<-WSN_Assembl
       TC-Hyg-2-Sensor<-WSN_Assembl
       Hum-TC-Hyg-1-Sensor<-WSN_As
       Hum-Hyg-1-Sensor<-WSN_Assem
       WSN-L-M<-WSN_Assembly-L-M
       TC-A-L-2-Sensor<-WSN_Assembl
       TC-A-L-3-Sensor<-WSN_Assembl
       WSN-B-M<-WSN_Assembly-B-M
       TC-B1-Sensor<-WSN_Assembly-E
       Hum-Temp-B1-Sensor<-WSN_Ass
       dP-B1-Sensor<-WSN_Assembly-B
       Hum-Temp-C1-Sensor<-WSN_Ass
       WSN-E-T<-WSN_Assembly-E-T<
       dP-D1-Sensor<-WSN_Assembly-E
       Hum-Temp-D1-Sensor<-WSN_Ass
       WSN-G-BB<-WSN_Assembly-G-I
       TC-Dust-G5-Sensor<-WSN_Assem
       dP-F1-Sensor<-WSN_Assembly-G
       ExTC-A1-Sensor<-WSN_Assembl
       ExTC-E1-Sensor<-WSN_Assembly
       ExTC-H1-Sensor<-WSN_Assembl
       TC-F2-Sensor<-WSN_Assembly-C
       TC-G3-Sensor<-WSN_Assembly-C
       ExTC-G1-Sensor<-WSN_Assembl
       WSN-G-BT<-WSN_Assembly-G-I
       TC-G4-Sensor<-WSN_Assembly-C
       TC-G2-Sensor<-WSN_Assembly-C
       TC-G1-Sensor<-WSN_Assembly-C
       TC-G5-Sensor<-WSN_Assembly-C
       TC-E3-Sensor<-WSN_Assembly-C
       TC-A3-Sensor<-WSN_Assembly-C
       WSN-A-T<-WSN_Assembly-A-T<
       TC-A2-Sensor<-WSN_Assembly-A
       TC-A4-Sensor<-WSN_Assembly-A
       TC-A1-Sensor<-WSN_Assembly-A
       TC-A5-Sensor<-WSN_Assembly-A
       ExTC-B1-Sensor<-WSN_Assembly
       TC-H2-Sensor<-WSN_Assembly-A
       TC-H1-Sensor<-WSN_Assembly-A
       TC-B2-Sensor<-WSN_Assembly-A
       WSN-C-M<-WSN_Assembly-C-M
       TC-C2-Sensor<-WSN_Assembly-C
       TC-C3-Sensor<-WSN_Assembly-C
       TC-D2-Sensor<-WSN_Assembly-C
```

Figure 7. ACAWS reacting to real-time failure in Mode: normal. The actual failure (28 VDC converter) produces ambiguity group consisting of the 28 VDC converter and 24 VDC converter. Since all ACAWS effect codes in the Status bar have been selected except for "LOR", chiclets bearing a functional relationship to the members of the ambiguity group are coded as effects that belong either to the common set or the set of "may be impacted".

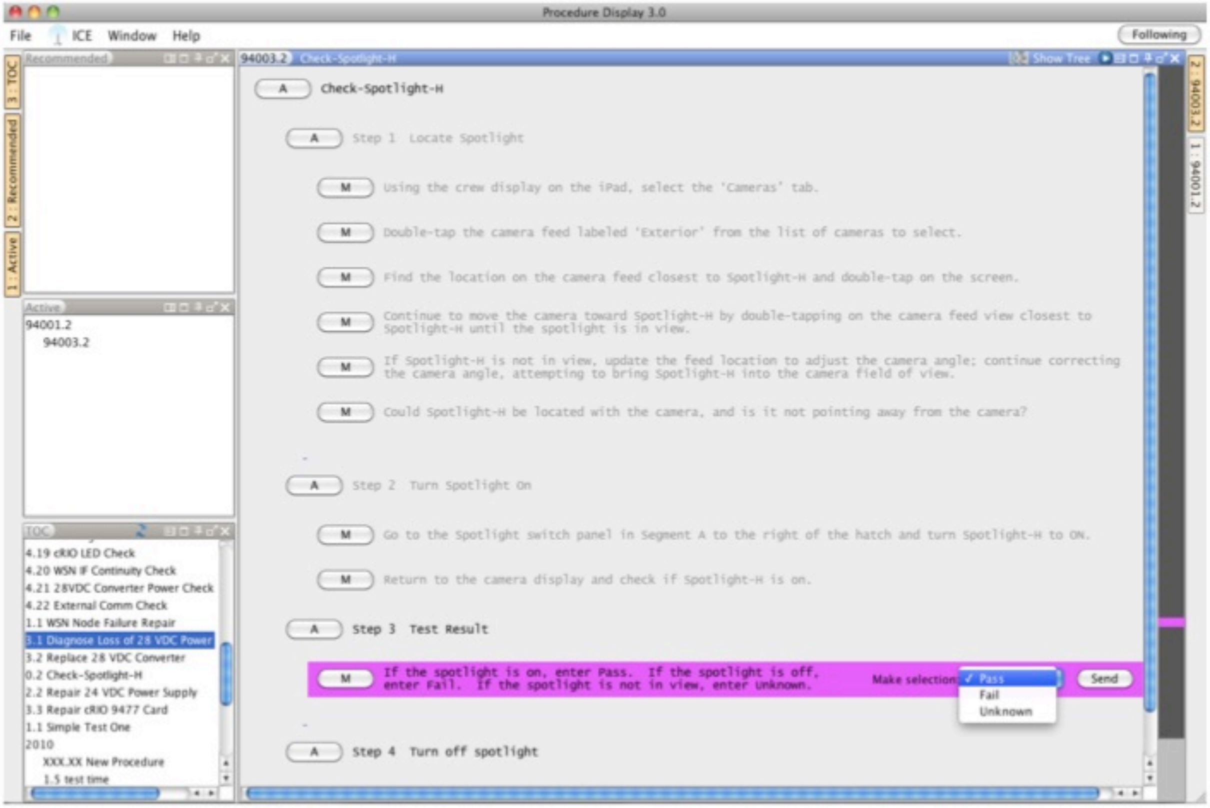
example, could help guide decisions as to what forms of replacement equipment might have the most "bang for the buck" when it comes to buying down mission risk, and making sure that the equipment is on board before departure. Loss of redundancy information could be used to answer a flight director's query on how quickly the flight control team needs to take action on troubleshooting to an unambiguous diagnosis, and many other potential queries.

D. Real-Time Mission Operations.

The ACAWS interfaces were also designed to support actual mission operations, particularly real-time fault diagnosis, isolation, and recovery. Let's illustrate the benefits by assuming that our formerly novice flight controller has completed her training,, and is now sitting on console during an actual mission. A Caution and Warning event occurs in connection with the DSH EPS system. Quickly reconfiguring the display to support analysis mode, what our controller sees is depicted in Fig. 7. Once again, ACAWS is unable to unambiguously diagnose the failure, returning instead an ambiguity group consisting of the 28 VDC and 24 VDC converters. In the status bar, all codes are selected except LOR, so all impacts are coded as either common set effects or "may be impacted" effects.

What does the flight controller do now? The first step is to see if ACAWS has suggested a possible procedure that includes manual tests that would provide additional information (test points) to disambiguate the failures. In the case of a 28 VDC and 24 VDC ambiguity set, there is such a procedure called "Check-Spotlight-H". This procedure, which our controller views on a dedicated procedure viewer (Fig. 8), contains a sequence of steps for the crew to carry out: Train a camera on an external spotlight, manually command the spotlight "ON", check the camera view to see whether the command was successful, and enter the test result through the procedure viewer ("Test Pass" if the light was on, "Test Fail" if not). Since ACAWS "knows" that the spotlight is only powered through the 28 VDC converter, once the operator inputs "Test FAIL" through the viewer, the model has the test result it needs to exonerate the 24 VDC converter as the source of the failure and declare the 28 VDC as the culprit.

On the ACAWS displays, the results of this disambiguation are shown in Fig. 9. Embedded in the 28 VDC chiclet(s) is the ACAWS failure marker, the yellow triangle with the embedded red X. All common and "may be impacted" effect codes have been removed, leaving only a clear indication, with "effect" coding, of what



94001.2
94003.2

4.19 cRIO LED Check
4.20 WSN IF Continuity Check
4.21 28VDC Converter Power Check
4.22 External Comm Check
1.1 WSN Node Failure Repair
3.1 Diagnose Loss of 28 VDC Power
3.2 Replace 28 VDC Converter
0.2 Check-Spotlight-H
2.2 Repair 24 VDC Power Supply
3.3 Repair cRIO 9477 Card
1.1 Simple Test One
2010
XXX.XX New Procedure
1.5 test time

A Check-Spotlight-H

A Step 1 Locate Spotlight

M Using the crew display on the iPad, select the 'Cameras' tab.

M Double-tap the camera feed labeled 'Exterior' from the list of cameras to select.

M Find the location on the camera feed closest to Spotlight-H and double-tap on the screen.

M Continue to move the camera toward Spotlight-H by double-tapping on the camera feed view closest to Spotlight-H until the spotlight is in view.

M If Spotlight-H is not in view, update the feed location to adjust the camera angle; continue correcting the camera angle, attempting to bring Spotlight-H into the camera field of view.

M Could Spotlight-H be located with the camera, and is it not pointing away from the camera?

A Step 2 Turn Spotlight On

M Go to the Spotlight switch panel in Segment A to the right of the hatch and turn Spotlight-H to ON.

M Return to the camera display and check if Spotlight-H is on.

A Step 3 Test Result

M If the spotlight is on, enter Pass. If the spotlight is off, enter Fail. If the spotlight is not in view, enter Unknown.

A Step 4 Turn off spotlight

Figure 8. Candidate Procedure Display showing steps to disambiguate the failure situation depicted in Figure 7. User has completed all steps above the magenta "focus bar" and is about to choose "Fail" from the pulldown menu of options for the results of the manual "turn spotlight on" step. This information is passed to ACAWS, which then has sufficient test point information to disambiguate the situation and declare the 28 VDC converter failed.

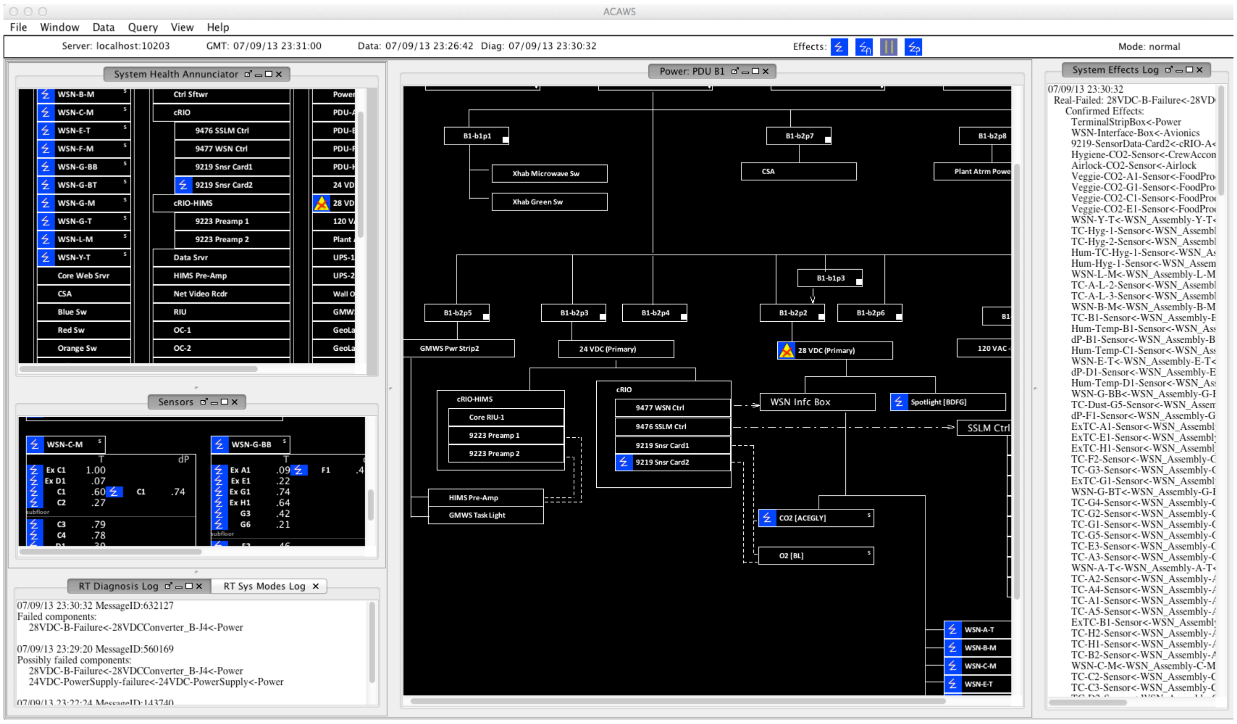
"downstream" components have been impacted as a result of the 28 VDC failure. Our flight controller is thus in a good position to proceed with mission planning and replanning steps in response to the failure.

III. Next Steps

Integrating ACAWS with even more external sources of information, data mining, and development of additional wrapper logic around the TEAMS model all offer avenues for adding even more capabilities to future versions of ACAWS. In this section, we discuss some of these enhancements, some of which are in work now, some scheduled for development in the future.

A. Next Worst Failure Determinations

When a spacecraft system experiences a malfunction,, the "landscape" of operational vulnerabilities and mission risks changes, sometimes dramatically. Following a failure, therefore, one of the highest priorities for flight controllers is to determine the "Next Worst Failure" (NWF), the failure that would have the biggest impact on mission goals and crew safety in the new context. The ACAWS ability to determine and display LOR status of failed parents and their impacted children is an important form of information that controllers bring to bear on NWF analyses. For example, the shuttles had two Freon loops (a primary and a backup) that transported excess heat generated by a wide variety of flight-critical components to radiators that dispersed the heat into space. Flight rules dictated an immediate de-orbit if the vehicle suffered a malfunction to the primary Freon loop, necessitating a switch



```
07/09/13 23:30:32
  Real-Failed: 28VDC-B-Failure<-28VD
    Confirmed Effects:
      TerminalStripBox<-Power
      WSN-Interface-Box<-Avionics
      9219-SensorData-Card2<-cRIO-A<
      Hygiene-CO2-Sensor<-CrewAccon
      Airlock-CO2-Sensor<-Airlock
      Veggie-CO2-A1-Sensor<-FoodPro
      Veggie-CO2-G1-Sensor<-FoodPro
      Veggie-CO2-C1-Sensor<-FoodPro
      Veggie-CO2-E1-Sensor<-FoodPro
      WSN-Y-T<-WSN_Assembly-Y-T<
      TC-Hyg-1-Sensor<-WSN_Assembl
      TC-Hyg-2-Sensor<-WSN_Assembl
      Hum-TC-Hyg-1-Sensor<-WSN_As
      Hum-Hyg-1-Sensor<-WSN_Assem
      WSN-L-M<-WSN_Assembly-L-M
      TC-A-L-2-Sensor<-WSN_Assembl
      TC-A-L-3-Sensor<-WSN_Assembl
      WSN-B-M<-WSN_Assembly-B-M
      TC-B1-Sensor<-WSN_Assembly-E
      Hum-Temp-B1-Sensor<-WSN_Ass
      dP-B1-Sensor<-WSN_Assembly-B
      Hum-Temp-C1-Sensor<-WSN_Ass
      WSN-E-T<-WSN_Assembly-E-T<
      dP-D1-Sensor<-WSN_Assembly-E
      Hum-Temp-D1-Sensor<-WSN_Ass
      WSN-G-BB<-WSN_Assembly-G-I
      TC-Dust-G5-Sensor<-WSN_Assem
      dP-F1-Sensor<-WSN_Assembly-G
      ExTC-A1-Sensor<-WSN_Assembl
      ExTC-E1-Sensor<-WSN_Assembl
      ExTC-H1-Sensor<-WSN_Assembl
      TC-F2-Sensor<-WSN_Assembly-C
      TC-G3-Sensor<-WSN_Assembly-C
      ExTC-G1-Sensor<-WSN_Assembl
      WSN-G-BT<-WSN_Assembly-G-I
      TC-G4-Sensor<-WSN_Assembly-C
      TC-G2-Sensor<-WSN_Assembly-C
      TC-G1-Sensor<-WSN_Assembly-C
      TC-G5-Sensor<-WSN_Assembly-C
      TC-E3-Sensor<-WSN_Assembly-C
      TC-A3-Sensor<-WSN_Assembly-C
      WSN-A-T<-WSN_Assembly-A-T<
      TC-A2-Sensor<-WSN_Assembly-A
      TC-A4-Sensor<-WSN_Assembly-A
      TC-A1-Sensor<-WSN_Assembly-A
      TC-A5-Sensor<-WSN_Assembly-A
      ExTC-B1-Sensor<-WSN_Assembl
      TC-H2-Sensor<-WSN_Assembly-A
      TC-H1-Sensor<-WSN_Assembly-A
      TC-B2-Sensor<-WSN_Assembly-A
      WSN-C-M<-WSN_Assembly-C-M
      TC-C2-Sensor<-WSN_Assembly-C
      TC-C3-Sensor<-WSN_Assembly-C
```

```
07/09/13 23:30:32 MessageID:632127
Failed components:
  28VDC-B-Failure<-28VDCConverter_B-J4<-Power

07/09/13 23:29:20 MessageID:560169
Possibly failed components:
  28VDC-B-Failure<-28VDCConverter_B-J4<-Power
  24VDC-PowerSupply-failure<-24VDC-PowerSupply<-Power

07/09/13 23:22:24 MessageID:143740
```

Figure 9. ACAWS display in Mode: normal following disambiguation information from manual test. The 28 VDC converter has been declared "failed" and all ACAWS coding associated with an ambiguity group have been removed. See text for more details.

to the backup. Such an event almost occurred on two shuttle missions when Freon flow through the primary loop fell to levels low enough to trigger hours of analyses to determine whether a switch to the backup was necessary. Had the switch actually occurred, ECLSS experts would have concentrated their NWF analyses on failures that would have caused the backup loop to also lose function.

However, LOR is only one consideration in making a NWF determination. Phase-of-flight considerations also play a central role. Imagine that, just as a de-orbit burn was imminent, the propulsion system responsible for the burn experienced a critical component failure, one of a large set of impacts of an electrical short in a high-level power bus. Although switching to a backup bus restores propulsion system functionality quickly, until the burn is complete, NWF determinations are going to skew heavily toward identifying components that would "take out" the backup bus, should they themselves happen to fail. Now suppose that the bus suffers the same malfunction a little later in the mission, after the burn has completed successfully, and the propulsion system in question has no more operative requirements through mission completion. The focus of NWF determinations would change to other effects of the bus short, particularly those that involve systems or equipment that figure in the remaining mission operations. ACAWS will require additional logic to incorporate this kind of dynamic knowledge. In less drastic cases, where the failures do not have loss-of-mission implications, considerations of impacts to mission productivity come into play. Mission controllers work with crews every day to develop detailed crew activity schedules that maximize crew productivity and best achieve mission operations goals. These activities often require equipment (e.g., drills, lights, vacuum cleaners, cameras, etc.) whose operation is dependent on systems resources such as power or data transmission lines, resources than can be disrupted (and hence, the activity halted) in the event of a systems malfunction. Under these circumstances, NWF considerations might take into consideration the impact of a failure on the current (daily) timeline of mission activities.

As a first step in that direction, we have recently added a preliminary version of an activity effects ("ActEffects") module to ACAWS. ActEffects links the existing failure isolation and consequence determination capabilities of ACAWS to two NASA mission activity generation and management tools, the Scheduling and Planning System for Exploration (SPIFe) and the Extensible Universal Remote Operations Planning Architecture (EUROPA). As shown in Fig. 10, the linkage allows us to inject a failure into the DSH EPS, determine and display crew activities on the activities timeline that are impacted by the failure, and automatically determine and schedule isolation and recovery procedures. ACAWS has diagnosed a failure in the DSH 24 VDC converter (the component that was exonerated in

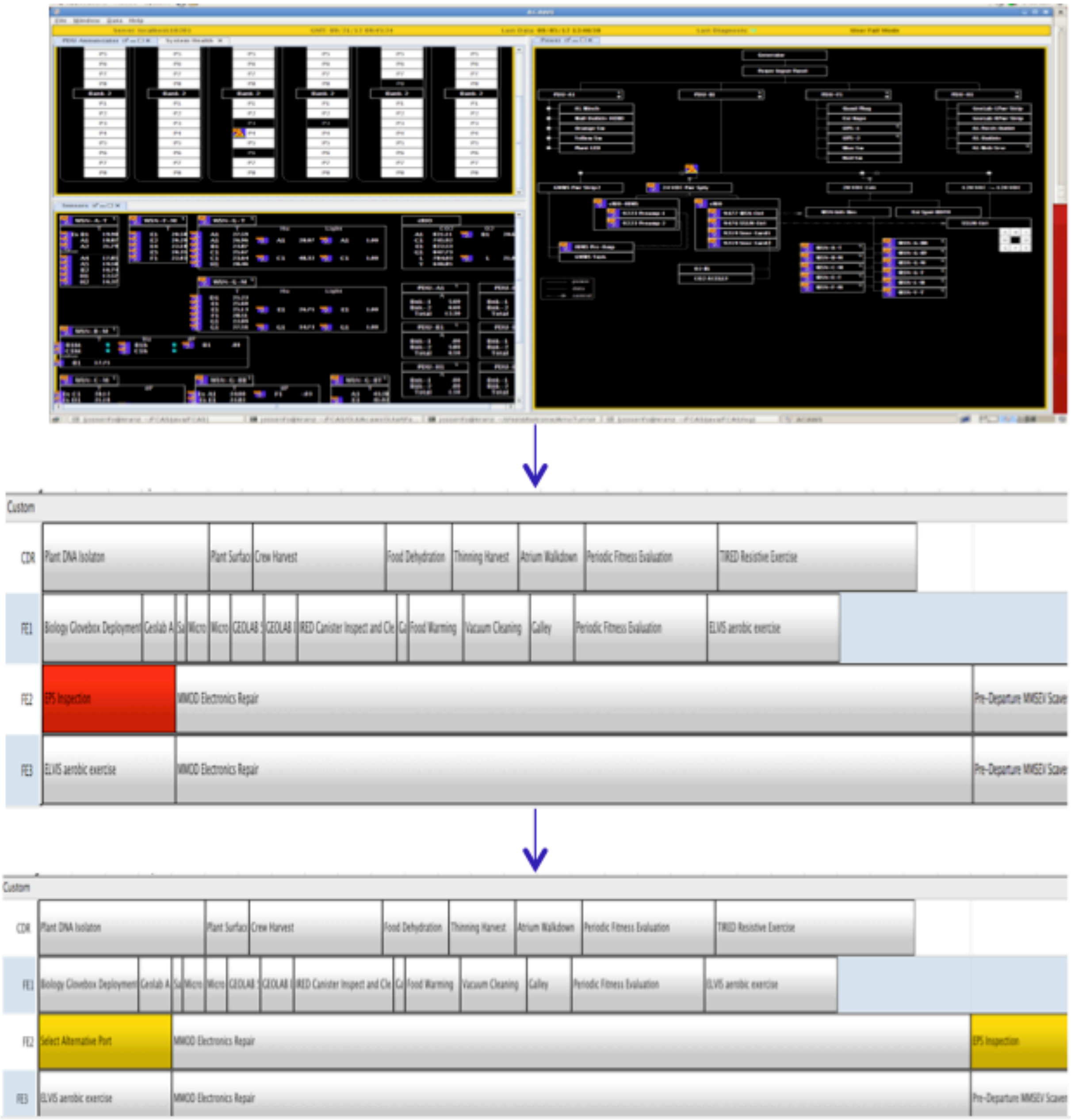


Figure 10. Example coordination between ACAWS and Planning and Scheduling Tools. See text for details.

our earlier example). The 24 VDC converter failure, or one or more of its effects, have disabled resources needed for crewmember "Flight Engineer 2" (FE2) to complete an EPS inspection activity at the beginning of his upcoming eight-hour work period. The disrupted activity is shown in the middle panel of the figure highlighted in red. As flight rules show that this activity is high priority, operators have determined that they want to schedule an immediate recovery procedure and reschedule the EPS inspection for later in the day. ACAWS automatically recommended a recovery procedure (Select Alternate Port) and provided a time estimate to complete it. SPIFe then automatically inserted the procedure as a new activity and rescheduled the EPS inspection as that crewmember's last activity of the day. The newly inserted activity and the rescheduled activity are shown in the bottom panel highlighted in gold.

B. Command and Control Coding

Existing TEAMS models represent system components largely in terms of functional dependencies, such as, if X powers Y, Y is dependent on X. Controllers, however, view systems components from a more operations-centric perspective. One of the functional dimensions of components that controllers care about is whether or not a component is commandable. Coding and representing such user-centered functional considerations in the TEAMS model would allow operators to explore more elaborate "what-if" failure scenarios that included impact assessments on individual controller roles, responsibilities, and actions.

C. Dynamic Procedure Authoring and Display Generation

Some of the most challenging real-time mission operations involve real-time procedure authoring in response to "unknown unknowns", the most dangerous class of failures that produce failure indications that not even human subject matter experts can make sense of initially, and require new procedures to be generated "on the fly" to "safe" the vehicle and investigate the source of the problem. Even at their current level of development, the ACAWS tools could play a useful role in assisting crewmembers or ground controllers in developing and working these contingency procedures, particularly when it comes to modifying existing procedures to take into account off-nominal operational modes and ongoing system (re)configurations. Although this area represents quite a challenging addition to ACAWS capabilities, a good argument can be made that crewmembers will require automated assistance with procedure authoring on deep-space missions where real-time ground assistance is unavailable. Similarly, providing automated assistance with understanding novel failure modes and unexpected impacts may well require dynamic generation of customized displays that show functional inter-system connections between components that are only represented today deep within paper-based systems schematics. The systems models that support ACAWS capabilities today could be used to extract such connections.

D. Natural Language Interfaces

Natural language interfaces with computer-based databases and other forms of computer-based knowledge representation systems are coming into their own in a wide variety of computing devices such as smart phones. An ability to interact with ACAWS via voice commanding and natural language based responding is a very attractive option for next-generation missions for model-based querying (e.g., "Is component X in the common set"? or "what is the next worst failure given failure Y") and procedure commanding. Again, natural language interfaces with ACAWS may become a requirement for deep-space missions to enable crewmembers to handle multi-tasking situations where their visual channels are fully saturated and they have no real-time ground assistance.

IV. Conclusion

The set of engineering products that flight controllers use to manage the health of onboard spacecraft systems today are not well integrated, impacting situation awareness, workload, and decision-making capabilities of controllers and crewmembers alike. Utilizing information technologies developed in the emerging ISHM field, we have developed an Advanced Caution and Warning System that consolidates the information and capabilities scattered across current engineering products into a single set of integrated information displays. This approach holds great promise for enhancing the situation awareness and decision-making capabilities of operators enough to support the stringent health management requirements of next-generation crewed missions to deep-space destinations.

Acknowledgments

We thank Ann Patterson-Hine, Mark Schwabacher, and NASA's Game Changing Development Autonomous Systems project for continued support of ACAWS development. We thank Gordon Aaseng, Vijay Baskaran, Adam Campbell, Silvano Colombano, Michael Dalal, Minh Do, Chuck Fry, David Iverson, Charles Lee, Paul Morris, John Ossenfort, and Brandon Oubre for vital programming and design assistance. The collaboration with the NASA JSC DSH team, NASA JSC flight controllers and trainers, and QSI's TEAMS team has also been invaluable.

References

[1]Frank, J, Spirkovska, L., McCann, R. S., Wang, L., Pohlkamp, K., and Morin, L., "Autonomous Mission Operations," *IEEE Aerospace Conference*, Big Sky, Montana, 2013.

[2]Johnson, S. B., Gormley, T. H., Kessler, S. S., Mott, C. D., Patterson-Hine, A., Reichard, K. M., and Scandura, P. H., (ed.), *System Health Management With Aerospace Applications*, Wiley, West Sussex, United Kingdom, 2011.

[3] McCann, R. S., and Spirkovska, L., "Human Factors", *System Health Management With Aerospace Applications*, edited by S. B., Johnson, T. H. Gormley, S. S. Kessler, C. D. Mott, A. Patterson-Hine, K. M. Reichard, and P. A. Scandura, Wiley, West Sussex, United Kingdom, 2011, pp. 319-338.

[4] Isherman, R., and Balle, P., "Trends in the Application of Model-Based Fault Detection and Diagnosis of Technical Processes", *Control Engineering Practice*, Vol. 5, 1997, pp. 709- 718.

[5] Muscettola, N., Nayak, P., Pell, B., and Williams, B., "Remote Agent: To Boldy Go where no AI System has Gone Before", *Artificial Intelligence*, Vol. 103, pp. 5-97.

[6] Narasimhan, S., and Biswas, G., "Model-based Diagnosis of Hybrid Systems", *IEEE Transactions on Systems, Man, and Cybernetics*, *Part A*, Vol. 37, 2007, pp. 348-361.

[7] Tumer, I. Y., and Stone, R. B., "Mapping Function to Failure During High-Risk Component Development", *Research in Engineering Design*, Vol. 14, 2003, pp. 25-33.

[8] Deb, S., Pattipati, K., and Shrestha, R., "QSI's Integrated Diagnostics Toolset", *Proceedings of the IEEE Autotestcon*, Anaheim, CA, 1997, pp. 408-421.

[9] Gill, T. R., Merbitz, J. C., Kennedy, K. J., Toups, L., Tri, T. O., Howe, A., S., and Smitherman, D., "Integration Process for the Habitat Demonstration Unit Deep Space Habitat", *Proceedings of the AIAA Space 2011 Conference and Exposition*, Long Beach, CA, 2011.