Developing Decision Aids to Enable Human Spaceflight Autonomy

Jeremy D. Frank, Kerry McGuire, Haifa R. Moses, Jerri Stephenson

■ *As NASA explores destinations beyond the moon, the distance between Earth and spacecraft will increase communication delays between astronauts and the Mission Control Center (MCC). Today, astronauts coordinate with MCC to request assistance and await approval to perform tasks. Many of these coordination tasks require multiple exchanges of information, (for example, taking turns). In the presence of long communication delays, the length of time between turns may lead to inefficiency or increased mission risk. Future astronauts will need software-based decision aids to enable them to work autonomously from the Mission Control Center. These tools require the appropriate combination of mission operations functions, for example, automated planning and fault management, troubleshooting recommendations, easy-to-access information, and just-in-time training. Ensuring that these elements are properly designed and integrated requires an integrated human factors approach. This article describes a recent demonstration of autonomous mission operations using a novel software-based decision aid on board the International Space Station. We describe how this new technology changes the way astronauts coordinate with MCC, and how the lessons learned from these early demonstrations will enable the operational autonomy needed to ensure astronauts can safely journey to Mars, and beyond.*

Historically, NASA's crewed missions have been limited to the Earth-moon system or low Earth orbit (LEO). Close proximity to Earth allows instantaneous communications between NASA's Mission Control Center (MCC) and astronauts on board the spacecraft. As NASA prepares for missions beyond the moon, including missions to Mars (the Evolvable Mars Campaign), the distance from the spacecraft to Earth will result in long communication delays. Table 1 shows the one-way light-time delay between Earth and some future mission destinations (Frank et al. 2015).

Currently, astronauts on board the International Space Station (ISS) depend upon the Earth-based Mission Control Center for technical assistance, troubleshooting, and daily schedule updates. If a piece of equipment fails on the ISS, MCC will be notified by the crew or by the data being streamed to the ground. MCC uses this information to evaluate the failure and decide how to fix the problem. Once a plan of action is determined, MCC provides instructions to the crew. This concept of operations is not sustainable for a crew at Mars; even simple problems may take minutes, or even hours, to resolve, due to the time delays shown in table 1. In order to

  ISSN 0738-4602

Destination	Distance (kilometers)	One-Way Time Delay (minutes)
ISS	435	ε
Lunar	38,400,000	0.02
Mars (close)	545,000,000	3
Mars (opposition)	4,013,000,000	22.3

Table 1. One-Way Time Delay Between Earth and Future Mission Destinations.

resolve this issue, astronauts will need to be autonomous, or operationally independent, from the Mission Control Center. Future crews will need the capability to evaluate system health independently and respond to failures. Doing so requires the ability to monitor system and vehicle health, troubleshoot issues, and maintain their daily schedules, all without relying on MCC. A higher degree of crew autonomy represents a fundamental change to mission operations. Enabling this new operations philosophy requires a host of operations protocol and technology development.

Decision Aids to Enable Autonomy

The problem of coordination between the Mission Control Center and the astronaut crew has some interesting features. The crew has the most up-to-date and detailed information on the state of the spacecraft, but MCC has the most in-depth knowledge and reasoning power, in the form of a larger team and computing resources. This imbalance leads to the need for coordination. Both parties must exchange information, over a small communication channel, in order to successfully perform the mission. This scenario is not unique to human spaceflight; similar problems arise in autonomous operation of piloted aircraft, as well as robotic control of spacecraft (for example, Mars rovers or orbiters), and other Earth-based autonomous vehicles. The introduction of autonomy-enabling decision aids changes a two-party coordination environment into a three-party environment: the human operator and decision aid take turns in a tightly coupled manner, with a looser coupling between the autonomous system and a distant mission control function over a small communication pipe.

NASA's interest in such decision aids includes human piloted aircraft as well as spacecraft. Examples of developments in this area include collision detection and avoidance (Kochenderfer, Holland, and Chryssanthacopoulos 2012), flight path situational awareness in modern glass cockpit aircraft (Neville and Dey 2012), emergency landing planning (Meuleau et al. 2011), and autonomous loss of control cueing and recovery of piloted aircraft (Klyde et al. 2014). While future human exploration of space will include highly dynamic piloting tasks, even seemingly mundane living and working tasks (for example, maintenance, repairs, science operations) require significant assistance from MCC, and can benefit from decision aids that enable astronaut autonomy.

A number of crew autonomy demonstrations have been performed in space, on board the ISS. These include crew autonomous procedures (astronauts performed numerous human spaceflight procedures without assistance from MCC, resulting in guidelines on the writing of these procedures [Beisert et al. 2013]); procedure automation (automating procedures normally performed by MCC, using software capabilities to both perform the task and notify astronauts — and MCC — of procedure execution status [Stetson et al. 2015]); and crew self-scheduling (a short demonstration to evaluate tools for astronauts to schedule their own daily activities).

Each of these demonstrations showed how to reduce the amount of coordination and communication between astronauts and MCC. Procedures can be written to include more information and reduce questions that arise during procedure execution. In cases where procedures or tasks require sending commands and receiving data from computerized systems, they can be automated (essentially by writing software to perform the tasks). Finally, crew self-scheduling reduces the need for astronauts to coordinate with MCC to schedule their own activities. However, these demonstrations did not require astronauts to take on the task of managing a spacecraft without the assistance of MCC.

The Autonomous Mission Operations Demonstration: An Overview

For seven months during 2014 and 2015, NASA's Autonomous Mission Operations (AMO) project demonstrated a decision aid on-board ISS using automated planning, fault detection and diagnostic technologies with failure response recommendations, easy-to-access references, and just-in-time training. Astronauts managed multiple ISS systems, including the total organic carbon analyzer (TOCA), a water quality analyzer, and the network of noncritical station support computer (SSC) laptops. TOCA analyzes the quality of recycled water; doing so requires the crew to sample the ISS water supply multiple times a month. The crew also performs numerous maintenance activities on time scales ranging from biweekly to annually. The SSC network functions like an office computer network for the crew's use.

Today, both of these systems are managed from the Misson Control Center. During this demonstration, the crew was asked to plan future TOCA water analysis and maintenance activities, monitor recently performed TOCA activities to ensure they were performed properly, diagnose hardware faults, and make recommendations in response to any problems encountered. These activities were performed once or twice a week for seven months by several ISS crew members. Managing such systems are the kinds of living and working tasks future astronauts may need to perform autonomously during future missions. The remainder of this article focuses on the crew tasks associated with crew monitoring TOCA performance, diagnosing TOCA faults, and recommending fault responses. For a complete discussion of the AMO experiment, see Frank et al. (2015).

How Turn-Taking Currently Works on the International Space Station

Current ISS operations are conducted with significant reliance on ground monitoring, control, and planning capability. MCC depends on nearly continuous communication coverage with the ISS for voice, telemetry, commanding, and video transfer to minimize, or in some cases eliminate, the need for onboard crew intervention. Flight controllers use a variety of tools and displays to closely monitor ISS systems. These tools are highly specialized for the specific spacecraft system and function being supported.

Mission control is roughly divided into two functions. The first function is to monitor and command ISS systems, and communicate directly with the crew. This is a real-time, moment-to-moment activity. The second function is a combination of long-term planning, in-depth data analysis, and troubleshooting recommendations.

Prior to the AMO demonstration, ISS crews would perform a TOCA analysis as planned by MCC, after which TOCA hardware performance and water quality data would be downlinked to the ground for analysis, sometimes hours after the activity was completed. If TOCA malfunctioned, or if water quality was abnormal, the crew would ask MCC for recommendations. After planning and analysis, MCC would send a recommendation to the crew for next steps. This two-party turn-taking model is shown in figure 1. If there is a communication outage, transfer of data or notification of next steps is delayed. It is notable that the crew has little or no insight into the current state of either water quality or system faults until MCC has performed its analysis. This situation is typical of many ISS systems today.

How Autonomy Changes Turn-Taking

The presence of high time delay changes the story of who takes the first turn. If a problem is time critical, then the crew may not be able to wait for MCC if something goes wrong. The number of such cases grows as the time delay and length of communication outages grow. This is the key driver for autonomy; crews must be able to handle these situations on their own. However, crew workload and coordination difficulty also increase with time delay. Prior work in understanding the impact of time delay on human spaceflight mission operations includes a large number of studies performed in different analog environments. These are summarized in the papers by Rader et al. (2013) and Frank et al. (2013).

The AMO software changes how turn-taking works, as shown in figure 2, by analyzing and presenting the TOCA data for the crew in real time, before flight controllers see the data. The AMO software also informs the crew of any situations that require a response (for example, TOCA faults or off-nominal water quality), as well as the recommended responses to the situations in question. Thus, the AMO software takes the first turn by analyzing all data from TOCA, whether water quality or device performance, before presenting the current state to the crew. The crew then takes the second turn, by consulting the AMO software to determine whether any actions are needed. MCC takes the last turn. This stands in stark contrast to today's mode of operations, shown in figure 1, in which little or no analysis is done by the crew or software on board.

It is also notable that the AMO software takes on a combination of mission control roles. The description of the system, including schematics, part lists, images, and so on, is the type of the information that flight controllers in the MCC use when monitoring systems and communicating with the crew. The automated planning and fault management technology is representative of tools used to perform the planning and analysis functions in the MCC. All of this information must be packaged for use by a single crew person.

Designing the AMO System

When designing an autonomous system that will supplement the support the crew receives from MCC, it is essential that the crew can trust the software to

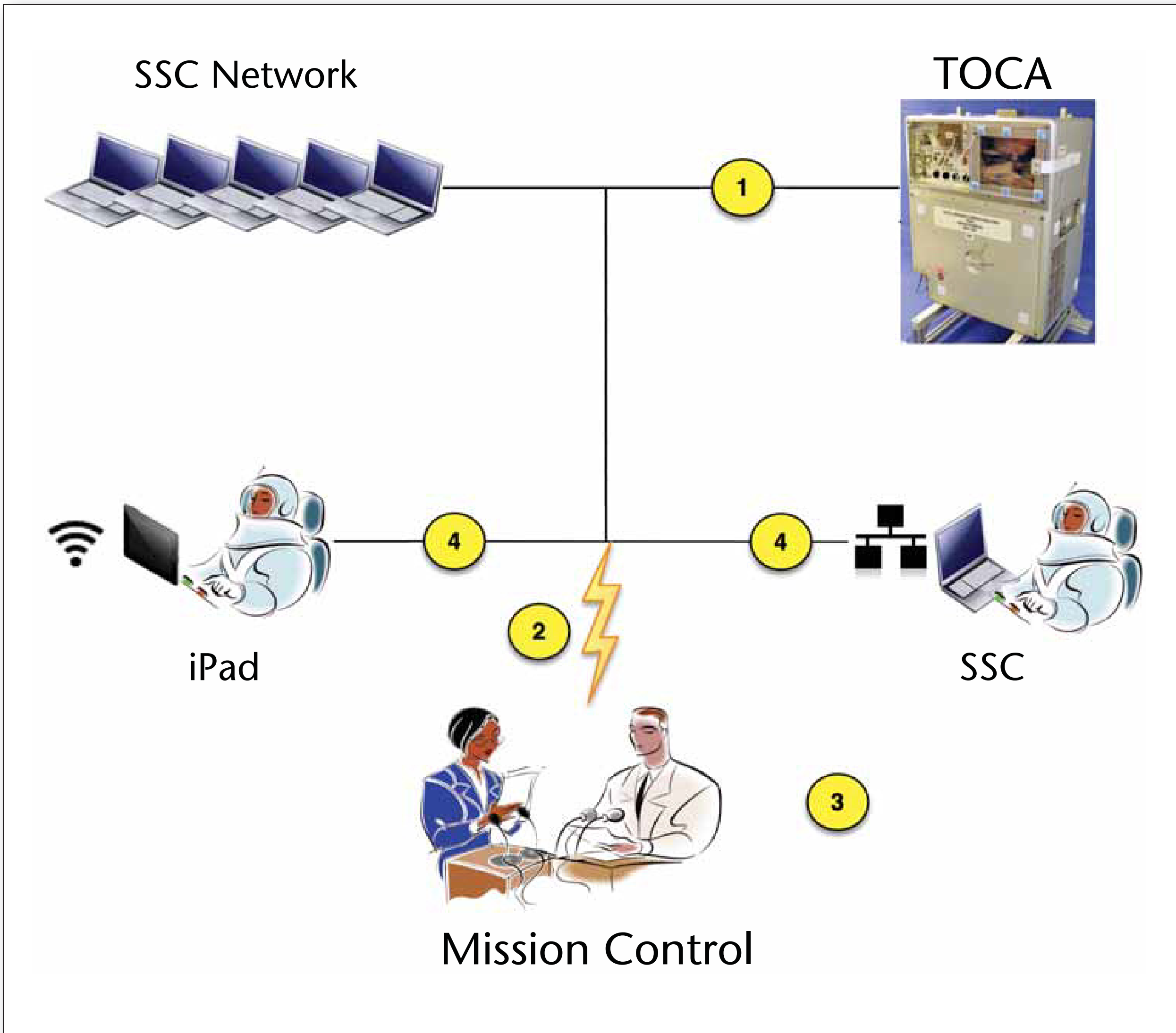

Figure 1. Operations on ISS Today.

(1) TOCA data is generated and (2) delivered to ground, often hours after TOCA is used. (3) The Mission Control Center analyzes the data and determines whether there is a problem, and what course of action, if any, is required, after which (4) the crew is notified.

lead them in the right direction. Previous research indicates that if human operators lack trust in the system, they will not use it (McGuirl and Sarter, 2006). On the contrary, when there is trust in the system being used, operators will be more reliant on it (Parasuraman, Sheridan, and Wickens 2008), and their performance may be superior to each (human or system) acting alone (Wickens, Gempler, and Morphew 2000). Appropriate allocation of tasks between the system and operator is required to achieve superior performance of operator and the system working together.

To achieve appropriate task allocation, a joint design process between flight controllers, software developers, human factors engineers, and crew was used to design and develop the AMO system. AMO team members conducted extensive research into how TOCA worked, as well as the ground scheduling and data analysis processes, to understand how the automated scheduling, fault detection and isolation, and anomaly detection technologies should be designed for use by the crew. Rapid prototyping and iterative design were emphasized in the user-centered development process. The involvement of subject matter experts (engineers who understood the systems) was necessary from the earliest software development phases to ensure accuracy, reliability, and usability.

The design of the user interface (UI) in an autonomous system is critical to ensure the users, in

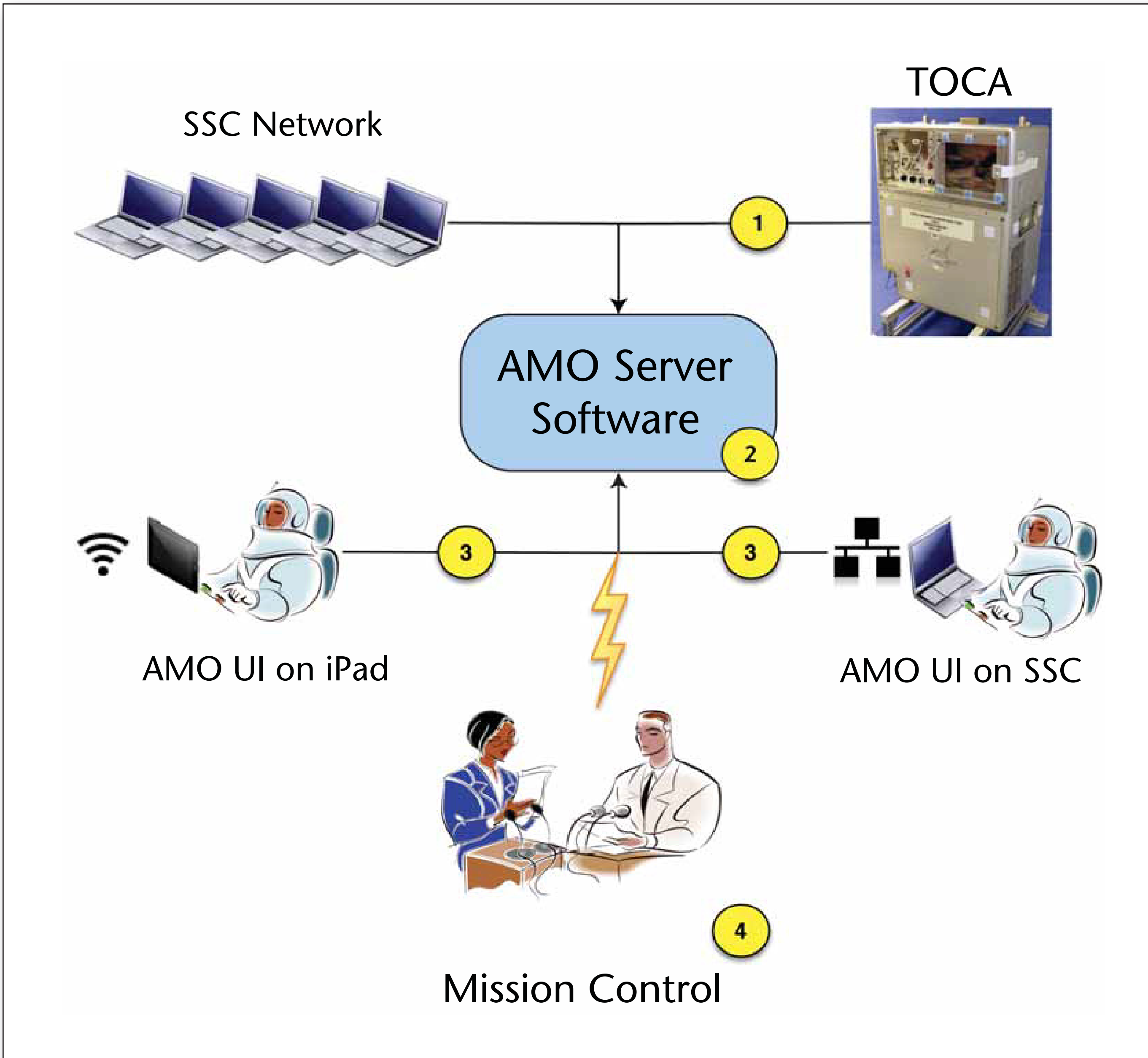

Figure 2. Taking Turns During an Off-Nominal TOCA Sample Analysis.

(1) TOCA data is delivered to the AMO software, which (2) analyzes the data in preparation for crew use. If crew members determine something is wrong (3) they notify MCC of the recommended action (4) and wait for confirmation

this case the crew members, have the appropriate level of information and insight into the system they are now taking turns with. The UI should provide information that can be understood with a quick glance. In particular, the UI should provide clear information regarding system state; is everything working properly, or is there a problem? If something is wrong, has a system failed, or is there an unexpected situation? With respect to recommended responses, the UI must clearly describe the recommended actions, the rationale for the recommendation, and the effects of the actions. Finally, the UI must provide means for the crew to provide information back to the system, in the form of manual data input or acknowledgement of actions performed. This turn-taking cycle between the crew and the software continues for the duration of the mission.

Turn-Taking and Coordination During the AMO Demonstration

The following example illustrates the new order of turn-taking using the AMO software in more detail. When testing ISS's water supply, the crew fill plastic bags with water for analysis by TOCA. If the bag is not completely filled with water, a condition known as a bag underfill, TOCA cannot analyze the water properly.

Before the AMO demonstration, communication for a prior bag underfill case lasted 4 minutes and 16 seconds over two separate communication sessions, with an elapsed time between communications of approximately 3 hours. Space-to-ground communication highlights the limited insight crew and ground had to the TOCA hardware before the AMO software:

Crew: Houston Space ground 2 for TOCA analysis
Ground: Go ahead
Crew: Ok we got a uh an alert that says analysis has been terminated fault detect on alert count two and uh I don't know what other information you need sounds like something didn't go right
…
Ground: Ok copy we're talking about it
Crew: And Houston on 2 for TOCA uh I'm looking at the uh sample bag and the sample bag is completely dry so there may have been not enough water in the sample bag
Ground: Ok copy that thank you
…
Ground: [crew] about a minute 10 'til [Loss of Signal] but if we you have the time we would like to know what those two alerts are uh you can pull those up just by hitting the acknowledge button once and then again for the second one
Crew: Ok stand by
Crew: Houston for TOCA. The first one is VCA move state timed out
Ground: Ok
Crew: And that's all the other one just says no message
Ground: All right copy thank you

During the 3 hours between communications, MCC conducted in-depth data analysis to come up with the next steps for the crew. On the way to Mars, the time to analyze the data may not change significantly, but the time to communicate varies as shown in Table 1, between 3 and 23 minutes. The space-to-ground communication snippets provided show the current back-and-forth style of communication that may last for an extended period of time. A simple problem that took a few hours to resolve may take a day or more on the way to Mars!

When a bag underfill occurred during the first use of AMO software on board ISS during the demonstration, the AMO software recognized the situation, and generated a notification for the crew. In this case, the crew member recognized the off-nominal message on the Results tab of the user interface, shown in figure 3. The crew member then successfully used the AMO software to diagnose the failure, and notified MCC with the correct recommended next action, which is to prime TOCA before performing the next sample analysis.

According to flight controllers, the communication during this failure was clear and concise (lasting only 2 minutes and 21 seconds) without the typical multiple back-and-forth questions between MCC and ISS. The following is a segment of the space-to-ground communication for this event.

Crew: I ended up with an error message on the TOCA read. And according to what I can figure out from my nice new app, the sample bag ran out of water. And so the idea here is I guess the next time you guys start the next TOCA run, it's recommended that TOCA be primed prior to next run.
Ground: [MCC] copies. All that's good information.
Crew: And I checked the bag, and it is dry.
Ground: And, [Crew], we'd like you to take the remaining steps per the error message software

After the demonstration, the crew member commented that the crew "understood the underfilled bag case immediately, all made sense to me. I looked at the bag, and sure enough the bag was empty; I knew exactly what was going on." The fact the crew was able to correctly diagnose the TOCA failure during its very first use of the software is a testament to the usability of the software.

AMO Demonstration Performance Analysis

To replicate the engineering analysis performed by ground teams, after each TOCA analysis the crew was asked to use AMO software to provide the ground with the following information: (1) Is the detected water quality trend nominal or off-nominal, and (2) Did TOCA hardware perform nominally or off-nominally during the run? If the answer to either question was "off-nominal," the crew was asked to provide a next-step recommendation to MCC. During the demonstration, crew was also directed to attempt to answer all TOCA-related questions using the software before calling MCC. Data was also obtained from crew and flight controllers on how the software was used, ease of use of the software, impacts to their situational awareness, and views on the change in roles and operations.

Crew and software both correctly classified 100 percent of the water quality trends as nominal or off-nominal. The crew and software correctly classified TOCA hardware behavior as nominal or off-nominal over 90 percent of the time. There were no false-positives; every time a TOCA error occurred, the AMO software indicated an error.

For more information about the remaining aspects of the demonstration, see Frank et al. (2015).

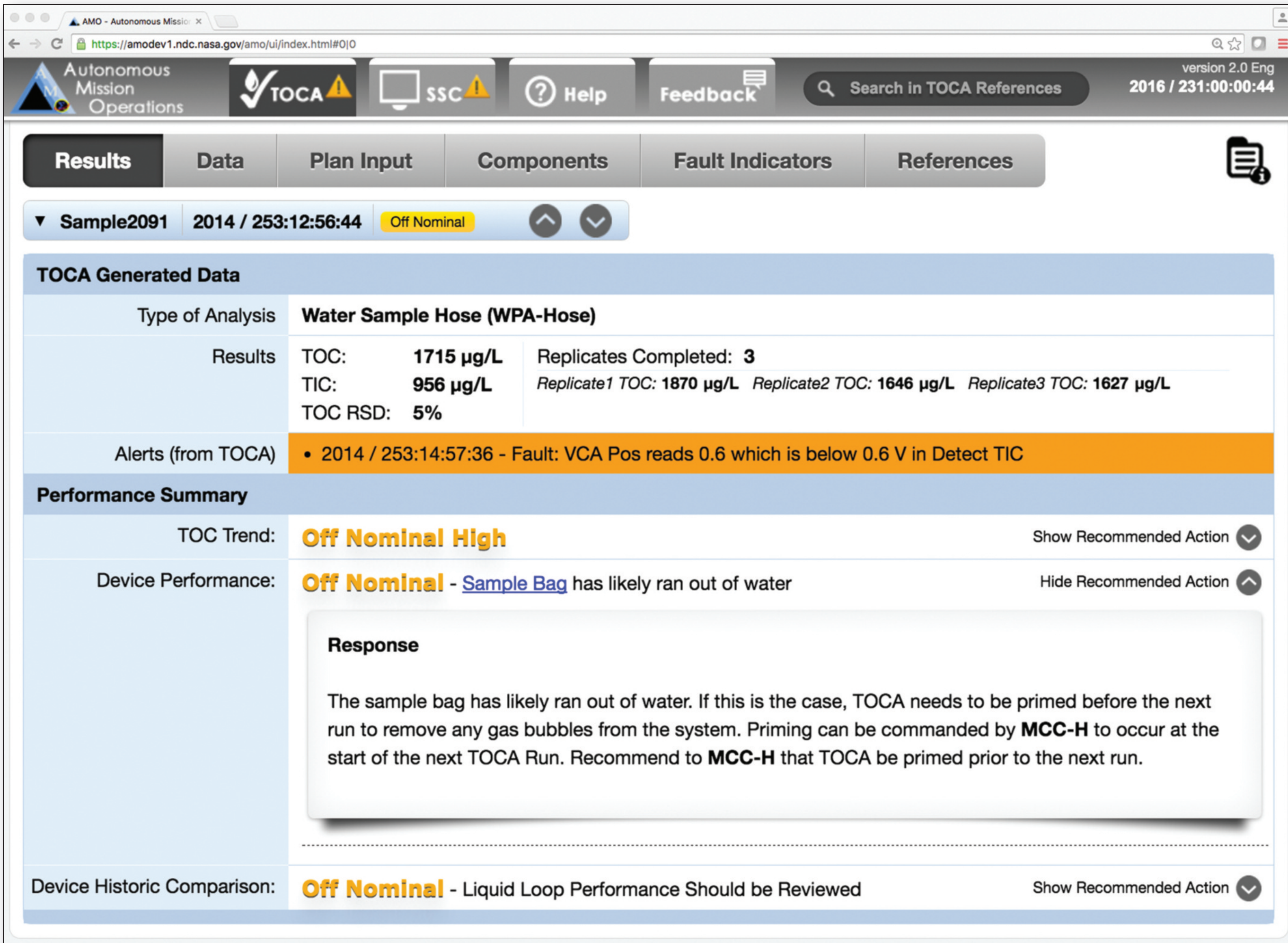

Figure 3. AMO Software.

The display shows an off-nominal message for a sample bag underfill with the recommended action for the crew.

Lessons Learned

AMO was the first demonstration of long-term crew autonomous management of a system on board a manned spacecraft. The AMO software allowed the crew to undertake increased responsibility for TOCA with no additional technical preflight training. The demonstration showed the crew was able to diagnose TOCA faults, and display a more detailed understanding of the system, when compared to occurrences of the same failures without AMO software assistance. Before the AMO software, water quality analysis could take hours to days. Now with AMO software, minutes after an analysis completes, the crew can analyze the results autonomously, and knows whether TOCA performed nominally. Also, the crew was able to use the software with only 30 minutes of preflight familiarization.

The software was well regarded by the astronauts; they trusted the information provided by the software, their situational awareness was increased, and they found the interface easy to use. Crew comments below illustrate these findings:

> AMO gave me everything I needed even when a very obscure error happened with TOCA.
>
> Typically — you'd get the "don't worry about that now" call from ground. With the software, I knew what was going on versus calling the ground.
>
> AMO software is beautiful and well organized.
>
> Incredibly easy and intuitive.

Crews put a lot of trust into the software recommendations, and were likely to follow the software's recommendations. A limitation with automation software is that it is only as good as the knowledge coded within it, and this knowledge is initially based upon preflight characterization of hardware performance. This limitation is likely to never be entirely overcome, and thus users should always be aware of the potential for errors based on incorrect or outdated knowledge. As understanding of system behavior improves, or in response to changes or degradation in performance, this knowledge must be updated if the software is to continue working properly. Software should allow easy updates to augment the preflight knowledge with knowledge gained during a mission; this will be especially important for long missions, as hardware and system performance changes due to changing operational environments or slow equipment degradation.

The benefits of these decision aids extended to MCC, whose role also changes in the presence of autonomy. Flight controllers mentioned that the AMO software helped diagnose potential issues before they even happened, and that they found the software useful in preparing for potential error messages. Flight controllers looked at data trends before crew call downs, viewed recommendations for error messages, and followed along with the crew. Even though the software had not been designed for flight controller needs, it increased their situational awareness. Including flight controllers in the design of the software from the beginning increased their acceptance and use of the software.

Despite the success of the demonstration and the positive feedback, numerous challenges remain in developing decision aids for autonomous systems. As noted in Frank et al. (2015), the AMO system did not perform perfectly, due to incomplete knowledge of all operating constraints and fault modes, and lack of access to all information on board ISS. This highlights several challenges. As noted, the astronaut crew trusted the system, but this trust must be balanced with a recognition that the system has limits; this challenge can only be addressed by a combination of testing, software design to ensure maintainability of knowledge, protocols for updating of system behavior, and adequate UI design and training to ensure that crew understand the system's limitations. A second set of challenges involves the difficulty of the automated reasoning needed to analyze system behavior and make recommendations. While the AMO system did not encounter these difficulties, as problems become larger and harder to solve, these challenges will become important to solve. A third challenge is in ensuring flexibility of decision support recommendations for complex situations. Other decision aids (Meuleau et al. 2011) are able to make several recommendations in response to a situation, and allow the crew to select the best; one such recommendation could be to "phone home." The AMO demonstration scenarios were simple enough that multiple recommendations were not required, but future scenarios may require trading off between options, including sitting tight and waiting for help from MCC.

Aiming High for Autonomy

As noted in the introduction, a three-party model for coordination and turn-taking is key to enable autonomy in a variety of contexts. These range from areas of interest to NASA, including piloted aircraft and future human spaceflight, but extend to other environments, including both surface ships and submersibles; mining; and exploration of remote or hostile environments. In all of these cases, decision aids may allow humans to operate complex systems under conditions in which communication with a remote planning entity are difficult or impossible to achieve.

The AMO demonstration showcased a new concept for human space flight operations that trades software development, informed by system experts, for preflight training, thereby allowing complex systems to be managed by non–subject matter experts. Throughout the demonstration, ISS astronauts' positive interactions and continued use showed their trust and acceptance of the software. The tasks and system used in this demonstration represent a small part of the operational needs of a future exploration spacecraft. Further demonstrations and experiments with ISS and ground analogs are needed to continue fleshing out autonomy enabling technologies, concepts of operations, system designs, and lessons learned. In the words of one astronaut who used the software "This is the way we ought to be heading. I know it is difficult to gather all that information, but if you could do it for other systems, it would be great!"

Acknowledgments

This work was funded by the NASA Advanced Exploration Systems (AES) program. The authors gratefully acknowledge the hard work and dedication of the large team of flight controllers, programmers, human factors engineers, and crew that made this demonstration possible.

References

Beisert, S.; Rodriggs, M.; Moreno, F.; Korth, D.; Gibson, S.; Lee, Y.; and Eagles, D. 2013. Development and Execution of Autonomous Procedures Onboard the International Space Station to Support the Next Phase of Human Space Exploration. Paper Presented at the 2013 AIAA Space Conference, 10–12 September, San Diego, CA.

Frank, J.; Iverson, D.; Knight, C.; Narasimhan, S.; Swanson, K.; Scott, M.; Windrem, M.; Pohlkmap, K.; Mauldin, J.; and McGuire, K. 2015. Demonstrating Autonomous Mission Operations Onboard the International Space Station. Paper presented at the 2015 AIAA Space 2015 Conference and Exposition. 31 August–2 September, Pasadena, CA.

Frank, J.; Spirkovska, L.; McCann, R.; Wang, L.; Pohlkamp, K.; and Morin, L. 2013. Autonomous Mission Operations. In *Proceedings of the 2013 IEEE Aerospace Conference*. Piscataway, NJ: Institute for Electrical and Electronics Engineers. dx.doi.org/10.1109/AERO.2013.6496927

Klyde, D.; Lampton, A.; Alvarez, D.; Richards, N.; and

Cogan, B. 2014. Flight Test Evaluation of the Safe-Cue System for Loss of Control Mitigation. Paper presented at the AIAA Science and Technology Forum and Exposition 2014: AIAA Guidance, Navigation, and Control Conference. 13–17 January, National Harbor, MD. dx.doi.org/10.2514/6.2014-0447

Kochenderfer, M.; Holland, J.; and Chryssanthacopoulos, J. 2012. Next Generation Airborne Collision Avoidance System. *Lincoln Laboratory Journal* 19(1): 17–33.

McGuirl, J., and Sarter, N. 2006. Supporting Trust Calibration and the Effective Use of Decision Aids by Presenting Dynamic System Confidence Information. *Human Factors: The Journal of The Human Factors and Ergonomics Society.* 48(4): 656–665. dx.doi.org/10.1518/001872006779166334

Meuleau, N.; Neukom, C.; Plaunt, C.; Smith, D.; and Smith, T. 2011. The Emergency Landing Planner Experiment. Paper Presented at the 2011 Scheduling and Planning Applications Workshop (Spark), in Conjuction with the International Conference on Automated Planning and Scheduling (ICAPS). 11 June, Freiburg, Germany.

Neville, R., and Dey, M. 2012. Innovative 787 Flight Deck Designed for Efficiency, Comfort, and Commonality. *Boeing Aero Magazine* 45(1).

Parasuraman, R.; Sheridan, T.; and Wickens, C. 2008. Situation Awareness, Mental Workload, and Trust in Automation: Viable, Empirically Supported Cognitive Engineering Constructs. *Journal of Cognitive Engineering and Decision Making* 2(2): 140–160. dx.doi.org/10.1518/155534308X284417

Rader, S.; Reagan, M.; Janoiko, B.; and Johnson, J. 2013. Human-in-the-Loop Operations over Time Delay: NASA Analog Missions Lessons Learned. Paper Presented at the 43rd International Conference on Environmental Systems (ICES). Vail, CO, 14–18 July.

Stetson, H.; Frank, J.; Cornelius, R.; Haddock, A.; Wang, L.; and Garner, L. 2015. Amo Express: A Command and Control Experiment for Crew Autonomy. Paper presented at the Annual International Space Station Research and Development Conference, 7–9 July, Boston, MA. NASA Technical Report 20150016490, Washington DC.

Wickens, C.; Gempler, K.; and Morphew, M. 2000. Workload and Reliability of Predictor Displays in Aircraft Traffic Avoidance. *Transportation Human Factors* 2(2): 99–126. dx.doi.org/10.1207/STHF0202_01

Jeremy Frank is the group lead of the Planning and Scheduling Group at NASA Ames Research Center. He received his Ph.D. from the Department of Computer Science at the University of California at Davis. He has a BA in mathematics from Pomona College. Frank's work involves the development of automated planning and scheduling systems for use in space mission operations, fault detection for space applications, and the development of technology to enable astronauts to autonomously operate spacecraft. Frank has published over 40 conference papers, six journal papers, and two book chapters, and received over 30 NASA awards.

Kerry McGuire is a space human factors engineer at NASA's Johnson Space Center (JSC). She obtained her BS in industrial engineering from Clemson University and her MS and Ph.D. in industrial and systems engineering from the University of Wisconsin – Madison. At NASA she divides her time leading human factors research to improve the usability of crew interfaces and increasing crew autonomy, mentoring students, and being the center lead for the Human Research Program Exploration Medical Capability Element.

Haifa Moses is a human factors engineer in the Human Systems Engineering and Development Division at NASA Johnson Space Center. She received her MS in human-computer interaction from the Georgia Institute of Technology and a BA in general psychology from Agnes Scott College. Her work at NASA includes investigating technologies from a user-centered perspective that will aid crew members on future long-duration missions such as wearable technology, advanced caution and warning for crew and ground, and crew displays for NASA's latest spacecraft Orion.

Jerri Stephenson is a human factors engineer in the Habitability and Human Factors Branch of the Human Systems Engineering and Development Division at NASA Johnson Space Center. She received her MA in applied cognitive psychology from University of Houston, Clear Lake and a BBA in both marketing and business management from Abilene Christian University. Her work at NASA includes evaluating innovative HCI technologies to assist crew on long-duration missions with a focus on smart habitats, virtual environments, and mobile computing.