Evaluating Spacecraft Crewstation Designs with a Simulator Driven by an AI Reactive Controller

Ujwala Ravinder[*], Peter A. Jarvis[+], Robert S. McCann, Robert Harris[+]
NASA Ames Research Center
Moffett Field, CA94035
{uravinder, pjarvis, rmccann, rharris,}@arc.nasa.gov

Abstract

Compared to today's missions, NASA's next generation of exploration mission will require new cockpit user interfaces and display formats that enable astronauts to operate their vehicles with less real-time assistance from the ground. To achieve a more autonomous concept of vehicle operations, the interfaces will have to ensure an optimal level of crew-vehicle interactions and minimize the possibility of crew error. This level of optimization is not guaranteed by the traditional approach to cockpit user interface display design and evaluation, which relies on building crewstation mockups and monitoring crew performance in high-fidelity simulations of crew-vehicle operations. This approach is too expensive and resource-intensive to permit more than a small number of design options to be evaluated. In this paper, we describe an ongoing program to develop a more automated evaluation approach via the integration of the Apex reactive planning system from the Artificial Intelligence community and a human performance modeling tool, the Man-Machine Integration Design and Analysis (MIDAS) simulation engine from the Human Factors community. MIDAS provides a comprehensive toolset for describing crewstation designs running simulated missions to determine an astronaut's expected performance using it. The system's procedural specification language is not sufficiently powerful to model the complexities of spaceflight rules. The Apex reactive controller provides the rich set of features required. We describe this integration and the significant utility of the application it enables. We present experimental results that show a high correlation between the system's predicted human performance and results obtained from physical simulations of fault detection, isolation, and recovery operations in ground-based simulations of shuttle ascents.

Keywords: Reactive Planner, Human Factors, Crewstation Design, and Simulation.

Introduction

The crews of today's spacecraft could not operate their vehicles without near-real time telemetry and communications links between the vehicle and the ground. However, these links are currently available only for missions in Low-Earth-Orbit (LEO). In sharp contrast, the goal of NASA's next generation of exploration vehicles is to carry astronauts safely to the "Moon, Mars, and Beyond". On such missions, the vehicles will be so distant that speed-of-light limitations will effectively eliminate near real-time links with the ground. Consequently, these missions are eventually going to require a more autonomous concept of vehicle operations in which the most time-critical activities, such as real-time fault detection, isolation, and recovery, will have to be performed on the vehicle itself. The operational paradigm has shifted from a "ground-centered" concept of vehicle operations to a "crew-centered" concept of vehicle operations.

A crew-centered concept of vehicle operations places new and stringent requirements for the design of crewstation displays, user interfaces, and choices for human-machine function allocation (levels of automation) choices. Crewstations will have to organize and present information to crewmembers in ways that are quick and easy to assimilate, that fully exploits human information processing capabilities and conform to human information processing limitations, and that support flexible forms of human-automation interaction.

Today's user interface and display technologies offer many new opportunities to optimize the efficiency and usability of crew-vehicle interfaces, compared to the current generation of crewed space vehicles. However, NASA's traditional approach to crewstation design is to build physical mock-ups and then monitor the performance of astronauts running through a series of simulated mission scenarios. Although this is a strong method, that provides extensive predictive data, it is expensive and demands significant amounts of time from a unique population (astronauts) who are heavily committed by their mission training requirements. It is not optimal to repeatedly push astronauts past the point of failure. These and other cost and risk factors limit the number of design options that can be explored with traditional methods to only a small number of the total possible designs, raising the risk that the most optimal design is not the one actually selected.

In this paper, we describe a new, more automated approach to evaluating crewstation design that holds considerable promise for streamlining the design/test/evaluate cycle and evaluate a much greater proportion of design options that can be done today. The approach integrates a human performance modeling (HPM) tool, the Man-Machine Integration Design and Analysis System (MIDAS) with the APEX AI reactive controller. MIDAS provides a comprehensive toolset for modeling a crewstation and an astronaut's interactions and performance within it. However, its procedure specification language is only capable of representing a single threaded, fully ground, and deterministic procedure. By contrast, spacecraft operations sometimes involve multiple malfunctions that must be handled concurrently and emergency checklists with multiple possible execution paths that must be navigated in real time. The simulation engine must also mitigate competition for resources especially in the case of situations, like these, that involve multiple competing demands for the crew's attention and processing resources. The Apex system provides these capabilities.

We structure this paper as follows. Section 2 introduces the space shuttle launch scenario that we will use to motivate the integration. Section 3 describes the current approach to evaluating designs with physical mockups. We include this primarily to show the level of instrumentation that is used to capture an astronaut's performance and therefore provide context to our experimental results. Section 4 describes our model-based approach to evaluating crewstation and procedure designs. We describe the MIDAS system and then the Apex system role in the application. Section 5 presents predictive results produced by the model and compares them against data from physical human-in-the-loop simulations using the same crewstation designs and procedures.

Scenario: Space Shuttle Ascent Phase

The ascent phase of a shuttle mission lasts approximately eight and a half minutes, and covers the period between liftoff and main engine cutoff (MECO) when the orbiter has reached its target orbit insertion point. Throughout this time the flight crew, composed of the mission commander and the pilot, are seated in the orbiter's cockpit (Figure 1), wearing pressure suits and experiencing up to three times the normal force of gravity. The shuttle flies on autopilot, with the flight crew continuously monitoring vehicle trajectory, abort options, preprogrammed activities such as solid rocket booster separation, and other critical systems through front and overhead displays. The original cockpit configuration contains three main cathode ray tubes for electronic information display (Figure 1) and the astronauts must frequently toggle controls to access the information they require.

Figure 2 provides a screen capture of an abort options display that was designed in a shuttle cockpit upgrade project that occurred between 1999 and 2004, but was not actually implemented. The vehicle's position is shown as a small circle overlaid on a map of the North Atlantic and a curve indicating the vehicle's course. The flight crew would have monitored this display to determine the options that are open in the event of a malfunction such as an engine failure. The options include return to launch site (land back at Kennedy Space Center) (RTLS), transatlantic landing (TAL) in Spain, and abort to orbit (ATO).

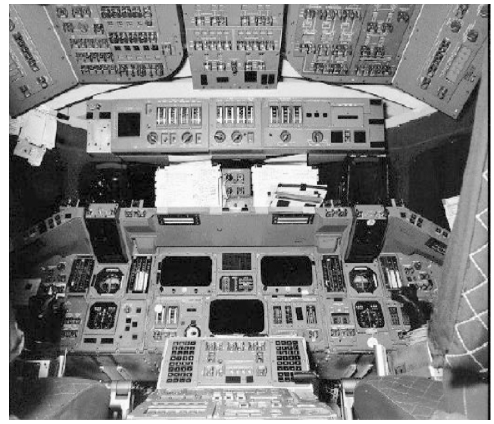


Figure 1: Orbiter Cockpit Crewstation

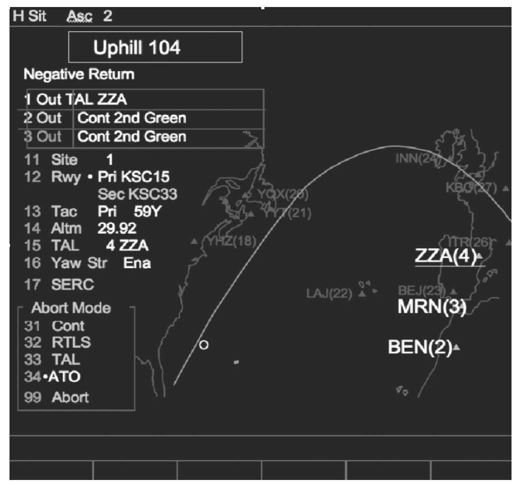
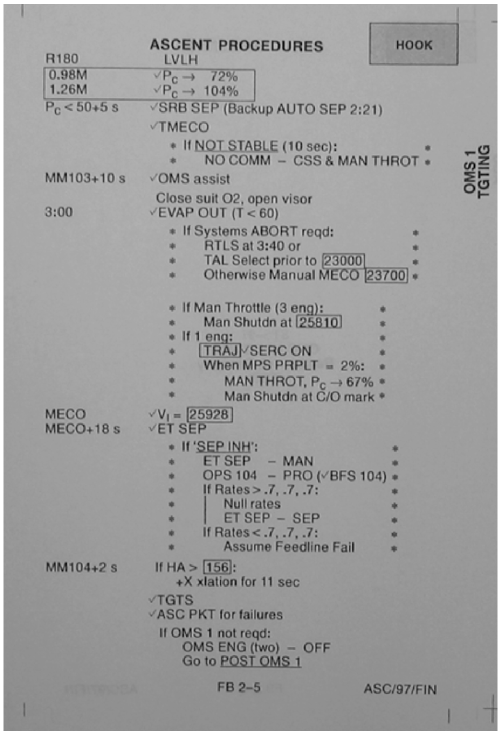


Figure 2: Orbiter Abort Options Display

Figure 3 shows an example of the orbiter's ascent flight procedures. Some flight procedures, such as the card in Figure 3, are velcroed around the cockpit on cue cards; others are bound into book form (an example is show on the pilot's left knee in Figure 4, which depicts an astronaut in a part-task shuttle cockpit simulator at NASA's Ames Research Center). In the nominal procedures list depicted in Figure 3, the left hand column indicates the point in the mission when the step described to the right applies. Note the *3:00* in the top third of the figure. This indicates that at a mission elapsed time of three minutes (time after launch) the crew must check that the Flash Evaporator system is functioning correctly by ensuring that a temperature reading associated the vehicle's Environmental Control and Life Support System is below 60 degrees and decreasing. If this system malfunctioned, the orbiter would start to overheat and possibly lead to an abort situation. The procedure specifies the abort steps after the *T< 60* condition. The actions are themselves gated. The first selects the appropriate mission abort option (RTLS, TAL). The second handles the case when another problem has occurred and the crew is controlling the main engines manually. The third condition covers the case when only one is engine is operating.



ASCENT PROCEDURES

HOOK

R180 LVLH

0.98M ✓$P_c \rightarrow$ 72%

1.26M ✓$P_c \rightarrow$ 104%

P_c < 50+5 s ✓SRB SEP (Backup AUTO SEP 2:21)

✓TMECO

• If NOT STABLE (10 sec):
• NO COMM – CSS & MAN THROT

MM103+10 s ✓OMS assist

Close suit O2, open visor

3:00 ✓EVAP OUT (T < 60)

• If Systems ABORT reqd:
• RTLS at 3:40 or
• TAL Select prior to 23000
• Otherwise Manual MECO 23700

• If Man Throttle (3 eng):
• Man Shutdn at 25810
• If 1 eng:
• TRAJ ✓SERC ON
• When MPS PRPLT = 2%:
• MAN THROT, $P_c \rightarrow$ 67%
• Man Shutdn at C/O mark

MECO ✓V_I = 25928

MECO+18 s ✓ET SEP

• If 'SEP INH':
• ET SEP – MAN
• OPS 104 – PRO (✓BFS 104)
• If Rates > .7, .7, .7:
• Null rates
• ET SEP – SEP
• If Rates < .7, .7, .7:
• Assume Feedline Fail

MM104+2 s If HA > 156:
+X xlation for 11 sec

✓TGTS

✓ASC PKT for failures

If OMS 1 not reqd:
OMS ENG (two) – OFF
Go to POST OMS 1

OMS 1 TGTING

FB 2–5 ASC/97/FIN

Figure 3: Page from Flight Procedures

Current Design Evaluation Approach

The facility shown in Figure 4 is called the Intelligent Spacecraft Interface Systems (ISIS) facility, and allows crewstation designs to be rapidly prototyped and evaluated (McCandless et al. 2005). Figure 4 shows the array of touch screens in the crewstation that allow different display designs and positions to be prototyped and integrated into simulations of shuttle ascents. The ISIS facility is heavily instrumented with eye trackers, cameras, audio recording, and software to time-stamp and log participant activities. This instrument suite allows researchers to precisely measure and evaluate participants' cockpit behavior in both nominal and off-nominal (i.e., following a systems malfunction) situations.

Typical design problems that motivate evaluation of display redesigns in the ISIS facility include displays that do not present systems information in a form that promotes rapid understanding of the nature of a systems malfunction, and physical arrangement of displays and switch panels that force the operator to look away from a critical display at a time when the display is providing critical information about the nature of the malfunction.

While the approach to new display evaluation in the ISIS simulator provides excellent quality evaluation data, it is expensive to operate. It demands precious operator time together with many hours of analysis to transform the raw data into a form that supports the design evaluation. The work reported in this paper was performed in collaboration with the ISIS team with the vision of enabling many more design options to be explored initially in software and with only the most promising being carried forward for examination in the physical simulator.

Figure 4: Astronaut Gregory H. Johnson in the ISIS laboratory

Integrated Apex and MIDAS Approach

MIDAS

The Man Machine Integration Design and Analysis System (MIDAS) (Corker & Smith 1993; Gore & Jarvis 2005) is the result of a fifteen-year multi million-dollar research program in the Human Factors Division at NASA Ames Research Center. The system provides the following facilities for modeling crewstations, flight control rules, and assessing the performance of the simulated operator.

Environment Model

The environment model represents the world outside of the vehicle. This model is not required for the shuttle launch phase because the crew is interested only in the operation of the vehicle. In other applications this model can be rich. For example, a military search and rescue mission would require a terrain model together with details of the objects (buildings, people, vehicles, etc.) within it.

Crewstation Model

The crewstation model represents each display and control within a crewstation. MIDAS provides a comprehensive set of displays and controls that can be specialized for a particular crewstation. Each display item is defined in terms of the information it displays and its precise position within the crewstation.

Anthropometric Model

The anthropometric model captures the physical constraints of the operator. Models range from petite females to large males. This range allows a design's effectiveness to be assessed on a full range of people. For example, a control may be found to only be reachable by a tall individual or a short individual may be discovered to have a better view on a particular display.

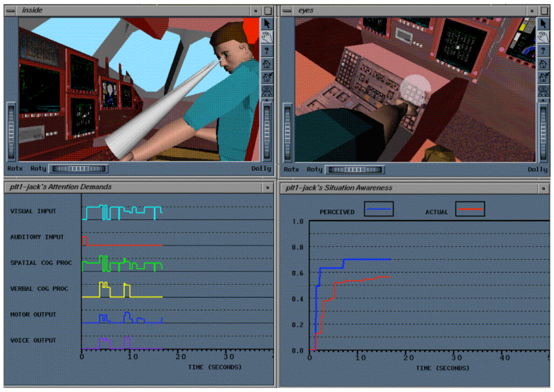


Figure 5: MIDAS Orbiter Crewstation Simulation

Native Procedure Specification Language

MIDAS' native procedure definition language (known as OPL) is a simplistic representation designed for specify a single deterministic procedure. It does not allow alternative methods for completing an operation to be specified or leave open the options for the resources that will be used to perform a task.

Visualization

Figure 5 presents a screen shot of MIDAS' visualization capability. It is primarily used to communicate a simulation's results by allowing users to observe the problems reported by the simulation

MIDAS is equipped with a sophisticated model of a users cognitive load. The graph on the bottom left of Figure 5 shows the loading on a user's visual and auditory input channels, mental spatial and verbal internal processing, and verbal and motor outputs. This model is based on Wicken's (1984) cognitive model. It is used to detect when a crewstation design and procedure places unreasonable loads on an operator. For example, it would register an problem when an astronaut is forced to try and reach for an awkward control lever while simultaneously having to read a display and listen to information coming from a radio.

The second graph on the right hand side of Figure 5 shows difference between the information available in the environment and the information the astronaut has observed. This gap is due to display changes that have occurred since and astronaut's last scan of it. Problems occur when the gap becomes large as a design forces an astronaut to be looking away from a display where critical information is changing.

MIDAS provides a sophisticated tool for describing a crewstation design and assessing an astronaut's load and performance while operating it. It's scripting language, however, is too simplistic to model the full range of flight procedures necessary to evaluate spacecraft operations. The language permits only a single deterministic execution thread where all parameters are bound ahead of time. Flight procedures demand concurrent execution where many binding decisions cannot be made until execution time together with the implementation of a prioritization strategy to mitigate competition for resources during busy periods such as multiple malfunctions occurring at the same time. These are precisely the set of feature supported by the Apex system.

Apex

Apex is a reactive controller developed at NASA Ames (Freed 1998) researchers. A member of the RAPs (Firby 1989) family of reactive controllers, APEX is an open source project and can be downloaded from the link given at the end of this paper. Apex is used extensively within NASA and outside on a broad range of applications. The most recent accomplishment was the successful control of an autonomous helicopter flying complex surveillance missions (Whalley et al 2005). The project's focus has been on providing a usable controller with an intuitive representation, comprehensive development tools, and extensive documentation. This *usable autonomy* focus proved critical to the successful application presented in this paper.

The core element of Apex is a reactive planning algorithm that selects actions based partly on a library of stored partial plans. Such planning algorithms are considered reactive because decisions about the next course of action evolve as new decision-relevant information becomes available. Apex synthesizes a course of action mainly by linking together elemental procedures expressed in Procedure Definition Language (PDL), a notation developed specifically for the planner. A PDL procedure consists of at least an index clause and one or more step clauses. The index uniquely identifies the procedure and describes a class of goals for which the procedure in intended. Each step describes a subtask or auxiliary activity presented by the procedure. Steps are not necessarily carried out in the order listed or even in a sequence. Instead, they are assumed to be concurrently executable unless otherwise specified.

Figure 6 presents the abbreviated encoding of the shuttle ascent phase procedure described earlier. Step *S0* states that a crewmember must scan the displays in the crewstation. The details of how to accomplish this will be specified in a separate PDL procedure with an index clause that unifies with this step. The *:repeating* clause requires that the scanning repeats throughout the lifetime of the ascent procedure at 20-second intervals. The *:priority* clause is used to mediate between competing procedure steps that are attempting to access a common resource.

Step *S10* calls a more detailed procedure for verifying that the flash evaporator defined back in Figure 3 is operating. The *waitfor* condition constraints when the Apex controller is to dispatch this task. In this case it must wait until mission elapsed time (MET) reaches three minutes. Step S11 is an example of a simple wait for constraint where it must wait until step S10 has terminated before it can proceed.

```
(procedure :concurrent
   (index ascent-phase)
   (step s0  (scan displays ?crew-member-1)
      (:repeating :with-min-interval '(20 secs))
      (:priority 10))
   …
   (step s10 (verify-flash-evaporator ?crew-member-2)
      (waitfor (MET = 3:00))
     (:priority 100))
   (step s11 (…)
      (waitfor s10))
   …
```

Figure 6: PDL Ascent Phase Procedure

Figure 7 presents the procedure that defines how to verify that the flash evaporator system in working. The index clause of the procedure unifies with the *S10* step in Figure 6. Step *S0* calls a more detailed procedure to have the astronaut examine the appropriate display to determine the current value of the critical temperature reading, and to study that display for ten seconds. Steps *S1* and *S2* encode the logic for determining if the temperature is within acceptable bounds. Step *S1* will report success if its *waitfor* clause is satisfied. This is a complex monitor *waitfor* which examines the value of the flash-evap temp variable for a period of ten seconds (*:duration)* at a rate of at least once a second (*:quality*). It succeeds if and only if the trend is monotonically decreasing and the value is below 60.

```
(procedure :concurrent
(index (verify-flash-evaporator ?crew-member))
 (step s0 (monitor flash-evap-temp ?crew-member))
 (step s1 (report flash-evap-temp nominal ?crew-member)
        (waitfor (:episode e1 (flash-ev temp)
                   :quality (:msi P1s)
                   :timing  (:duration (>= P10s))
                   :trend (:rate :decreasing)
                   :value (< 60))))
 (step s2 (handle flash-evaporator-failure ?crew-member)
        (waitfor (:episode e2 (flash-ev temp)
                   :quality (:msi P1s)
                   :timing  (:duration (>= P10s))
                   :trend (:rate :increasing)
                   :value (>= 60))))
 (step (terminate)
   (waitfor (:or ?s1 ?s2))))
```

Figure 7: PDL Verify Flash Evaporator Procedure

We must emphasize that the Apex's focus on usability has been as critical to its success in this application as its functionality. In this application with cognitive scientists we were able to author and debug shuttle flight procedures in PDL and then debug them using the Sherpa user interface.

Experimental Results

We compared the Apex/MIDAS system's predicated performance with the actual performance of several ISIS participants on physical mockups across two crewstation designs. The *Baseline* case corresponds to the current proposed shuttle orbiter cockpit design and *FAMSS* to an alternative, more advanced cockpit design.

Figure 8 shows the results obtained for scenario centered on the response to a simulated helium leak encountered during launch. Simulated results are prefixed by "model". The Apex/MIDAS system correctly predicts that the astronaut will perform better with the FAMSS design and the time lines are tightly correlated with those of the physical simulation. The large difference with the final step of the baseline procedure arose because the human subjects frequently overlooked this step and did not realize until much latter into the launch sequence at which point they would return and resolve it. Some did not return to resolve it. We do not have a good model of how humans over look things and therefore do not include it in our simulations. The human subjects took additional time to cross check displays in ways not specified in the flight procedures explain the remaining difference in times. We do not have a sufficient model of when a human will incorporate additional checking actions to include it within the simulation.

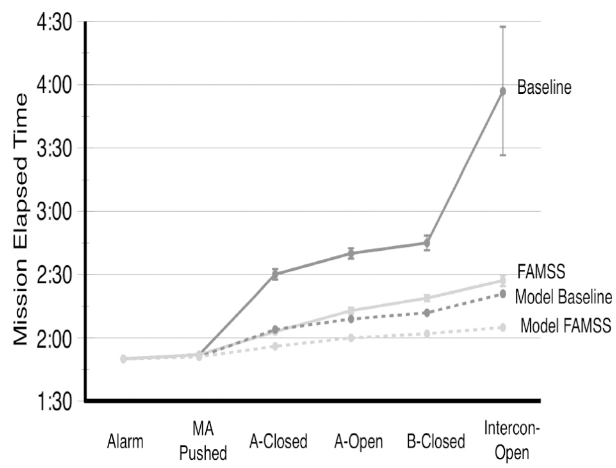


Figure 8: Isolatable Helium Link Malfunction

Figure 9 shows results obtained for another simulated malfunction. The integrated system correctly rates FAMSS as the better design. The human subject performing additional cross checking once more explains the difference in time.

We are confident given these results that the integrated system will correctly evaluate design options. We are in the process of adding additional features to the human performance model to increase its fidelity. We hope such models can express human behavior variation, human error, etc. The Apex/MIDAS integration has produced an effective tool for complementing physical simulations.

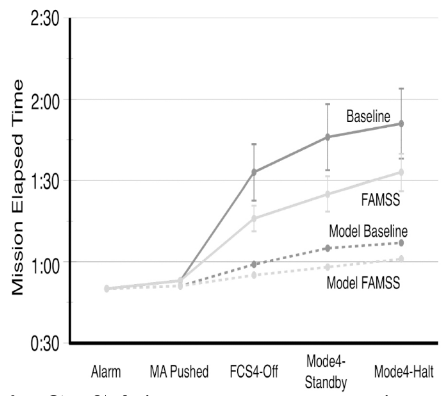


Figure 9: GPC failure to synchronize malfunction

Related Work

The AI community has produced a large number of reactive controllers. See Verma et al. (2005) for an excellent survey.

Georgeff and Ingrand (1990) report applying the PRS executive to the task of controlling the Space Shuttle system's orbiter's Reaction Control Engines (RCS). However, the models created by for PRS have been lost over time (Myers 2006) so we were unable to study them to aid in our model development.

Conclusion

We have reported on the integration of AI reactive controller with a Human Factors simulator to provide a powerful tool for evaluating crewstation and flight procedure designs. Our experimental results show that this approach is effective at predicting human performance on a design. The technology will allow NASA to evaluate a much broader range of designs as the agency moves forward to design the exploration vehicle required to meet it exploration goals of crewed missions back to the Moon and then on to Mars.

Apex is now the default controller for MIDAS and we are confident that this integration will lead to a large number of applications of the controller modeling procedures in domains as diverse as homeland security immigration processing and scientific experiment design on the International Space Station. These domains all involve humans working with complex machines through interfaces that must be carefully designed.

This application demonstrates the maturity of reactive controller technology and we hope that reporting on this application success encourages others to seek out opportunities to apply the technology. The controller used in this research is an open source system and freely available from the links given at the end of this paper.

References

Corker, K., and Smith, B. 1993. An architecture and model for cognitive engineering simulation analysis: Application to advanced aviation automation. In *Proceedings of the AIAA Computing in Aerospace 9 Conference*. October, 1993: San Diego, CA.

Firby, R., 1989. *Adaptive execution in complex dynamic worlds*. Ph.D. thesis, Yale University.

Freed, M. 1998. Managing Multiple Tasks in Complex Dynamic Environments. In *Proceedings of the 1998 National conference on Artificial Intelligence*, Madison, WI.

Georgeff, M., and Ingrand, F. 1990. Real-Time Reasoning: The Monitoring and Control of Spacecraft Systems. In *Proceedings of the Sixth IEEE conference on Artificial Intelligence Applications.*

Gore, B., and Jarvis, P. 2005. Modeling the complexities of human performance. *In Proceedings of the 2005 IEEE International Conference on Systems, Man, and Cybernetics*, The Big Island, Hawaii, USA

McCandless, J., Hilty, B., and McCann, R. 2005. New displays for the space shuttle crewstation. *Ergonomics In Design*, 13(4), 15-20.

Myers, K., 2006. Personal email communication.

Whalley, M., Freed, M., Harris, R., Takahashi, M., Schulein, G., and Howlett, J. 2005. Design, Integration, and Flight Test Results of an Autonomous Surveillance Helicopter. In *Proceedings of the AHS International Specialists' Meeting on Unmanned Rotorcraft*. Mesa, AZ.

Verma, V., Jónsson, A., Simmons, R., Estlin, T., and Levinson, R. 2005. Survey of Command Execution Systems for NASA Robots and Spacecraft. In *Proceedings of Plan Execution: A reality Check*. Workshop at the International Conference on Automated Planning and Scheduling Systems, Monterey, CA.

Wickens, C. 1984. *Engineering Psychology and Human Performance*. E. Merill Publishing Co, Columbus, OH.

Websites

Apex: http://ic.arc.nasa.gov/projects/apex/
MIDAS: http://human-factors.arc.nasa.gov/dev/www-midas/
ISIS laboratory: http://isis.arc.nasa.gov/

Acknowledgements

The authors would like to thank Edward Balaban, Brent Beutter, K. Mike Dalal, Brian Gore, Valerie Huemer, and Christian Neukom for their assistance with the work reported in this paper.