Procedure Automation: Sharing Work with Users

Debra Schreckenghost, Scott Bell, David Kortenkamp, and James Kramer
TRACLabs, 16969 N. Texas Ave, Suite 300, Webster, TX 77598
schreck@traclabs.com

Abstract

An area of interest for NASA is the use of procedures as the basis of task automation. The PRIDE software was developed to author and execute electronic procedures for NASA spacecraft and habitat operations. We describe our approach for modeling human-automation work based on a procedure language, and allocating and execution tasks among a human-automation team. We illustrate our approach with examples of collaborative work using procedure automation.

Procedure Automation for NASA

The PRIDE software was developed to author and execute electronic procedures for NASA spacecraft and habitat operations. The nature of work in NASA operations requires specialized knowledge about complex systems that may be used infrequently. Additionally, error consequences when performing the job can be significant. NASA uses procedures as a means of "planning ahead" how operators will perform both nominal and off-nominal work, to mitigate the risks of operating in such a high criticality domain.

Procedures are used to manage spacecraft and habitat systems, perform Extra Vehicular Activities (EVAs), and conduct space science and exploration. Astronauts and flight controllers are trained using procedures. Qualifying for flight control positions includes performance using procedures. Thus, NASA users are familiar with procedures and procedures are well-maintained.

NASA is interested is the use of procedures as the basis of task automation. As astronauts move deeper into space, their workload is expected to increase because Earth-based flight controllers will not be in continuous real-time communication. Task automation has potential to reduce astronaut workload for such missions. It also can improve response time as communication latency with Earth increases. And automation can prove beneficial in performing tasks prone to human error, such as vigilance monitoring.

One challenge in automating procedures is *capturing procedure knowledge* that can be used both for manual and automated execution. These task models often are built when the manual procedure is first documented, and well before the automation is available. Thus, our approach must produce electronic procedures for either manual or automated execution. NASA procedure authors are subject matter experts, so we also need an approach to task modeling that does not require computer programming skills.

Another challenge in automating procedures is *communicating automation behavior* and its effects on spacecraft and habitat systems. The introduction of task automation into NASA operations requires establishing operator trust that automation is reliable and predictable. Even when operating at a high level of automation, it is expected that operators must maintain awareness of automation actions, because they are responsible to direct and manage automation. It also is expected that operators will intervene when automation or system behavior is different than expected.

The PRIDE electronic procedure software was developed to address these challenges. It consists of a procedure editor (Pride Author), web-based display server (Pride View), and an automation engine (PAX). We describe our approach for modeling human-automation work based on a procedure language, and allocating and executing tasks among a human-automation team. We illustrate our approach with examples of collaborative work using procedure automation. We summarize our studies of performance with procedure automation. We propose to present our position with demonstration at the workshop.

Modeling Human Work for Automation

Inspection of the procedures used by NASA human space flight reveals an underlying action vocabulary and grammar for using this vocabulary that has a clearly defined semantics. When managing spacecraft or habitats, operators need to perform actions such as 1) send *commands* to a system 2) *verify* sensed values are as expected 3) *record* sensed values at a specific point in the procedure, and 4) *wait for* a sensed value to reach a target value. These atomic actions are composed into checklists with conditional action sequencing, such as 1) performing a subset of ac-

tions *conditional* upon situated information, and 2) *looping* through a subset of actions until a condition is true. PRIDE task models are represented using a procedure representation language (PRL; Kortenkamp, et al., 2008) that abstracts this vocabulary in a set of instruction types for building the action sequences seen in procedure checklists.

One user of PRIDE procedures is the procedure author who creates and modifies the PRL. For NASA, procedure authors are subject matter experts. They usually are engineers, scientists, or mathematicians. While they understand how to use a computer, they often have no background in computer programming. They typically use Microsoft Word to author procedure documents that are translated into XML files by a programmer. The Pride Author software provides a way for authors to produce XML directly while manipulating instruction objects (Izygon, et al 2008).

To add an instruction, the author drags the desired instruction type (corresponding to an action) into a central canvas area. This produces an instance of that type. Manipulation of these instruction objects in the canvas automatically produces PRL in the background. What the author sees is an action-object pair similar to what they typed into the Word document e.g., a valve enable command is displayed "Cmd CO2 Vent Valve Enable". The author also drags items from a model of the system commands and data (called the *System Representation*; Bell, et al., 2015) to insert references to system commands and telemetry verifies. Figure 1 shows an example of the procedure editing user interface for building PRL procedures.

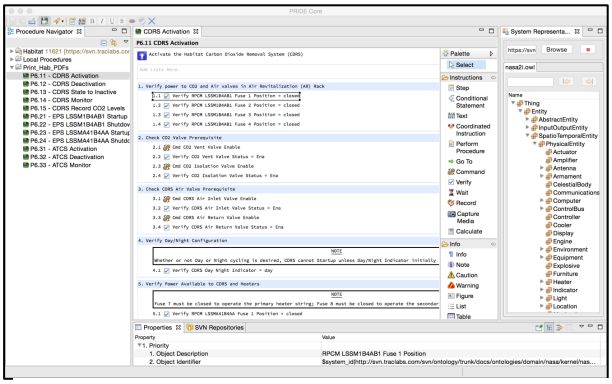

Figure 1. Pride Author User Interface

When executing the procedure, both the operator and automation use the same PRL task model to perform tasks. This model combines information to instruct a person what actions to take with information needed to execute those actions. Thus, a task to compare a sensed instrument reading to a target value will include both operator directions for what values to compare and data references for accessing current sensed readings. This model is used to generate a web user interface of the procedure document that is directly manipulated by a person to perform the task. The same model is used by the software to automate tasks.

As the procedure instructions are executed, the procedure display is annotated with information about the state of execution (what has been done, what remains to be done); see Figure 2. The same annotations are used whether a person or automation performs the task.

Thus, the same task-based user interface is used to monitor the actions of automation as is used to perform actions manually. This *shared task model* is the basis of human-automation communication about the task. Structuring the work of automation according to human work improves the transparency of automation actions. This approach provides a means for establishing common ground about the ongoing task that should improve operator understanding of automation behavior (Clark and Brennan, 1991).

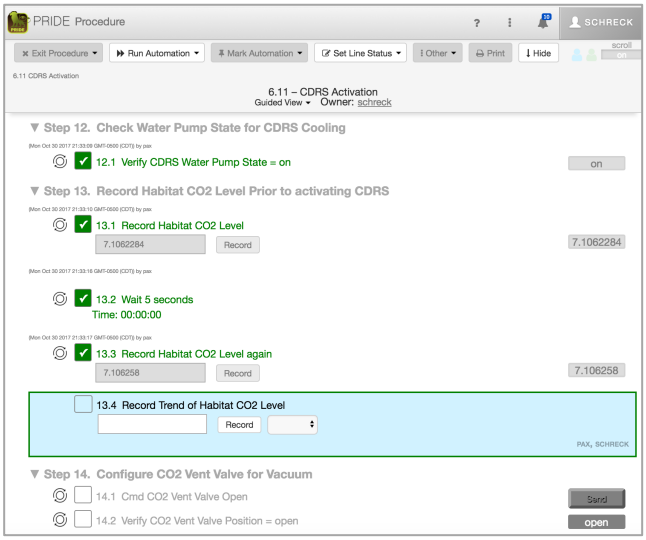

Figure 2. Pride View User Interface

Sharing Task Responsibility with Automation

Shared human-automation work for complex, high risk domains benefits from the ability to tailor the task allocations to the situation. For example, workload balancing may require a redistribution of tasks among the human-automation team. For electronic procedures, this means shifting or sharing the responsibility to perform instructions or make decisions between operators and automation. Each instruction is designated as manual only or automatable. *Manual only* instructions can only be performed by a person. *Automatable* instructions can be performed either by automation or a person. For the domains in which PRIDE procedures have been used, the ability to designate an instruction as *Automated Only* has not been needed. These designations are made when the procedure is au-

thored, and can be adjusted as needed when a procedure is performed (Schreckenghost, et al., 2008).

Responsibility to complete an instruction can be shared by the operator and automation. Instructions have an optional Witness property indicating when a person should approve the action taken by automation before proceeding to the next instruction. Failure of a human witness to approve the instruction is considered anomalous execution.

Procedure instructions are designed to be executed in the order shown in the procedure. When performing instructions manually, however, the user is able to alter the order of execution. PRIDE provides functionality (*oversight mode*) to alert the user when doing an instruction out of order, but such re-ordering is not prevented. When performing instructions automatically, the order of execution is enforced by the automation (*guided mode*). The currently "active" instruction is indicated by a colored, labeled focus bar placed behind the instruction. The operator can only manipulate command buttons or other interaction forms in the active instruction; all other instructions are disabled for manipulation until the focus bar reaches them.

Procedures can be composed of a mix of *Manual Only* and *Automatable* instructions. When operating in guided mode, the automation will pause when it reaches a *Manual Only* instruction. The interaction forms for that instruction are enabled for manipulation. If the user completes the manual action, the focus bar moves to the next instruction and automation resumes, if the instruction is designated *Automatable*. The user also has the option to skip the instruction, fail the instruction, or stop automation.

Examples of Collaboration with Automation

Multiple procedures can execute concurrently, operating at different levels of automation and with different types of human involvement. This supports a variety of human roles when performing collaborative work using procedure automation. We describe some examples of collaborative work with procedure automation below.

Joint human-automation work. Procedure instructions are executed by both the operator and the automation. Tasks are allocated according to policies, such as risk reduction. For example, some NASA operations rely on flight crew to assess the risk of issuing system commands and thus require all commands be sent by a person, while verifies and records can be done automatically. In other operations, human error may pose the greater risk and tasks will be allocated to automation. Allocations may be adjusted differently when executing the same procedure under different circumstance. For example, after changing out a sensor the operator may perform instructions manually that would normally be automated, to ensure that the new sensor behavior matches that expected in the procedure. Figure 2 shows an example of a joint human-automation procedure for starting up a Carbon Dioxide Removal System (Schreckenghost, et al., 2015).

Human supervision of automation. The operator decides which procedures to perform and when to perform them, while the automation executes most of the procedure instructions. Additionally, the human assesses whether automation performance is acceptable. Work design for this style of collaboration includes minimal operator performance of instructions, since the operator's primary responsibility is to manage the work. Often direct intervention by the person is an indication of work breakdown. An example of human supervision of automation is the use of procedure automation to manage the work of an autonomous robot. In one application of the PRIDE software, the operator assigns procedure sequences to a humanoid robot for the purpose of configuring switches.

Distributed human-automation teams. This type of collaboration requires users to perform coordinated work while physically distributed. Procedure automation represents another "team member" available to perform work. For example, all Extra Vehicular Activity (EVA) by NASA astronauts requires two astronauts working outside the vehicle and at least one crew member inside the vehicle or on Earth. For such work, multiple instruction sequences are ongoing concurrently. It is necessary to identify coordination points where these sequences must synchronize. PRIDE can designate instructions as "coordinated," which adds concurrency metadata used during execution. Specifically, it links two instructions in different procedures and identifies whether they should be performed simultaneously or serially. These metadata about coordination points should be respected by both humans and automation.

Performance with Procedure Automation

We have evaluated human performance using PRIDE automation in a number of NASA experiments. To establish a baseline for manual performance we compared manual use of PRIDE procedures with use of an analog for International Space Station (ISS) electronic procedures (Billman, et al., 2014). A key difference between these systems is that live data and commands are embedded in PRIDE procedure displays while data and commands are accessed from a separate display for ISS. Condition effects for both completion time and number of successful users were large enough to be significant for small n (11). Mean completion time was reduced by approximately half. No users had command errors using PRIDE while all users but one had command errors using ISS displays. Next, we compared manual use of PRIDE with PRIDE automation. Preliminary results indicate a reduction in execution timing and workload when using automation without a loss of situa-

tion awareness (n=27; Holden et al., 2018). We expect more performance improvement when users multi-task with procedure automation. We are investigating strategies for work allocation to improve performance when multi-tasking with automation.

Conclusions and Future Work

PRIDE automation is an example of a knowledge-based system using a hierarchical task language PRL to automate system monitoring and control. It includes rule-based activation of action sequences based on sensed data. Other similar systems include Reactive Action Packages (Firby, 1989), Task Description Language (Simmons, et al, 1998), and Plan Execution Interchange Language (Estlin et al., 2006). Unlike these systems, PRIDE was designed for humans and automation to perform shared procedural work, which requires effective human-automation communication and collaboration. The ability to designate tasks dynamically to either humans or automation is an example of a hybrid human-AI collaboration. Our user interface for procedure automation uses human task models to improve communication of AI behavior to users. All automation actions correspond to actions in human-comprehensible procedures, making these actions transparent and predictable, and potentially improving trust in automation.

While it is possible to reactively select which procedure to automate based on current conditions, PRIDE does not support reactively modifying procedure actions or action sequences. An area for future research is the use of machine learning techniques to adapt existing procedures or create new ones from task observations. Programs such as DARPA's Explainable AI (XAI) can provide techniques for learning procedural sequences that are more understandable and usable by users.

Our development of a procedure editor allowing subject matter experts to author executable procedural task models is an example of a tool for non-AI specialists to build AI models. An area for future research is adding constraint satisfaction tools to help non-AI specialists author procedures that respect domain action sequence constraints.

Finally, the current procedure user interface is intended for monitoring automation while performing low-level actions. For users to multi-task manual procedures with automated procedures, new user interface designs are needed that help users maintain automation awareness without vigilance monitoring of these low-level actions.

Acknowledgements

Human studies with PRIDE were performed with collaborators Dr. Dorrit Billman/ San Jose State University, and Dr. Kritina Holden/ Leidos. PRIDE technology was developed under NASA SBIR. Work on human supervision of automation was supported by the National Space Biomedical Research Institute through NASA NCC 9-58.

References

Bell, S., P. Bonasso, M. Boddy, D. Kortenkamp, and D. Schreckenghost (2015). PRONTOE: An Ontology Editor for Domain Experts. *Communications in Computer and Information Science*, Vol. 454. Fred, A., Dietz, J.L.G., Liu, K., Filipe, J. (Eds.)

Billman, D., D. Schreckenghost, & M. Pardis (2014). Assessment of Alternative Interfaces for Manual Commanding of Spacecraft Systems: Compatibility with Flexible Allocation Policies. *Human Factors and Ergonomics Society Annual Meeting*, Chicago.

Clark, H. H., & Brennan, S. E. (1991). Grounding in communication. In L. B. Resnick, J. M. Levine, & S. D. Teasley (Eds*.), Perspectives on socially shared cognition* (pp. 127--149). Washington, DC, USA: American Psychological Association.

Estlin, T, A. Jonsson, C. Pasareanu, R. Simmons, K. Tso, and V. Verma. Plan Execution Interchange Language (PLEXIL). *NASA Technical Report TM-2006-213483*. April 2006.

Firby J (1989) *Adaptive Execution in Complex Dynamic Domains*, Ph.D. Thesis, Yale Univ. Technical Report #672 Jan 1989.

Holden K. Schreckenghost D. Greene M. Hamblin C. Lancaster J. Morin L (2018). Electronic Procedures for Crewed Missions Beyond Low Earth Orbit (LEO). Presentation at *NASA HRP Investigators Workshop*, Galveston, TX, Jan 2018.

Michel Izygon, David Kortenkamp, and Arthur Molin (2008). A procedure integrated development environment for future spacecraft and habitats. In *Proceedings of the Space Technology and Applications International Forum (STAIF 2008)*.

Kortenkamp, D. R. P. Bonasso, D. Schreckenghost, K.M. Dalal, V. Verma, and L. Wang. (2008) A Procedure Representation Language for Human Spaceflight Operations, *9th International Symposium on AI, Robotics & Automation in Space, i-SAIRAS-08*

Schreckenghost, D., R. P. Bonasso, D. Kortenkamp, S. Bell, T. Milam, C. Thronesbery. (2008) Adjustable Autonomy with NASA Procedures. *9th International Symposium on Artificial Intelligence, Robotics and Automation in Space*, Pasadena, CA.

Schreckenghost, D., D. Billman, and T. Milam. Effectiveness of Strategies for Partial Automation of Electronic Procedures during NASA HERA Analog Missions (2015*). International Joint Conferences on Artificial Intelligence, Proceedings of AI in Space Workshop* July 2015. Buenos Aires, Argentina.

Simmons R. and Apfelbaum D. A Task Description Language for Robot Control, *Proceedings of Conference on Intelligent Robotics and Systems*, Vancouver Canada, October 1998.

Biography

Debra Schreckenghost is a Senior Scientist at TRACLabs. She conducts research in the areas of adjustable autonomy, human interaction with automation, and real-time performance of robots and automation. Scott Bell is the software lead for the PRIDE procedure system. Jim Kramer developed the PRIDE PAX software. David Kortenkamp is the TRACLabs' President and a key designer of PRIDE.