Talking through procedures: An intelligent Space Station procedure assistant

G. Aist[1], J. Dowding[1], B. A. Hockey[1], M. Rayner[1], J. Hieronymus[1], D. Bohus[2], B. Boven[3], N. Blaylock[4], E. Campana[4], S. Early[5], G. Gorrell[6], and S. Phan[7]

[1]RIACS/NASA Ames Research Center
{aist, jdowding, bahockey, mrayner, jimh}@riacs.edu

[2]Carnegie Mellon University
dbohus@cs.cmu.edu

[3]Kalamazoo College
bboven@acm.org

[4]University of Rochester
blaylock@cs.rochester.edu
ecampana@bcs.rochester.edu

[5]DeAnza College/NASA Ames Research Center
searly@mail.arc.nasa.gov

[6]Linköpings Universitet
gengo@ida.liu.se

[7]Santa Clara University
nphan@scudc.scu.edu

Abstract

We present a prototype system aimed at providing spoken dialogue support for complex procedures aboard the International Space Station. The system allows navigation one line at a time or in larger steps. Other user functions include issuing spoken corrections, requesting images and diagrams, recording voice notes and spoken alarms, and controlling audio volume.

1 Introduction

The International Space Station recently entered its second year as the first permanent human presence in space. Astronauts on board Station engage in a wide variety of tasks on orbit, including medical procedures, extravehicular activity (EVA), scientific payloads, and station repair and maintenance. These tasks are documented in the form of hierarchically organized procedures. In some cases, a procedure will be performed by one astronaut with another astronaut reading the procedure out loud; in other cases the astronaut will use the procedure and reference a paper (or onscreen) copy of the procedure. The RIALIST group has been developing a spoken dialogue system for providing assistance with Space Station procedures. This system has been developed in a cooperative, iterative endeavor with substantial input from astronauts, trainers, engineers, and other NASA personnel. The first version of the system operated on a simplified (and invented) procedure for unpacking and operating a digital camera (Aist et al. 2002), and included speech input and speech output only. In this paper we report on the current version of the checklist assistant as of December 2002, which is set up to run on XML-formatted actual Space Station procedures and includes speech input and multimodal output (speech, images, and display of HTML-formatted text.)

2 Motivation

The current crew on the ISS is limited to 3 astronauts. During pre-flight training, astronauts receive training on basic systems operation, and practice carrying out carefully designed procedures to handle both nominal and off-nominal operations. The number and variety of the procedures, as well as the duration of ISS missions,

precludes the kind of detailed training common to shorter Apollo and Shuttle missions. Astronauts on Station need to carry out procedures that they may not have trained on specifically in advance, or may not have practiced for a considerable time. Current practice may require the astronaut to follow through the procedure using a text or computer monitor, or to have a second astronaut read the procedure out loud to the one executing it.

Our approach is to develop a spoken dialogue system provide assistance in reading the procedure, tracking the progress through the procedure, and providing other assistance to support correct and complete execution. The dialogue system would thus free up the second astronaut for other tasks, increasing Space Station utilization.

3 System description

The fundamental architecture of the system consists of several components: audio processing, speech recognition, language understanding, dialogue management, HTML and language generation, and visual display and speech synthesis.

3.1 Audio processing, speech recognition

We use noise-canceling headset microphones for audio input, transmitted via Sennheiser wireless units to a laptop. Speech recognition is done with Nuance 8 using a context-free language model constructed from a unification grammar and then compiled into a recognition model (Dowding et al. 1993; Rayner, Dowding, and Hockey 2001).

3.2 Parsing and interpretation

The output of the speech recognizer is parsed using SRI's Gemini parser. The text of the recognized speech and the resulting parse are then fed to Alterf (Rayner and Hockey 2003), a robust interpretation module which combines statistical and rule-based interpretation to produce a sequence of tokens, such as "[load, water]". These tokens are then assembled into predicate-argument structure such as "load(water)".

3.3 Dialogue management

We adopt a TRIPS-style division (Allen, Ferguson, and Stent 2001) of dialogue management into three sections: input management, behavior management, and output management (Figure 1). In the December 2002 Checklist architecture, however, there are multiple behavior agents, each specialized by dialogue task: handling annotations (e.g. pictures and voice notes), manipulating system settings (e.g. volume), and handling procedure-based tasks (e.g. navigation). The dialogue input manager coordinates the interactions between the multiple behavior agents.[1]

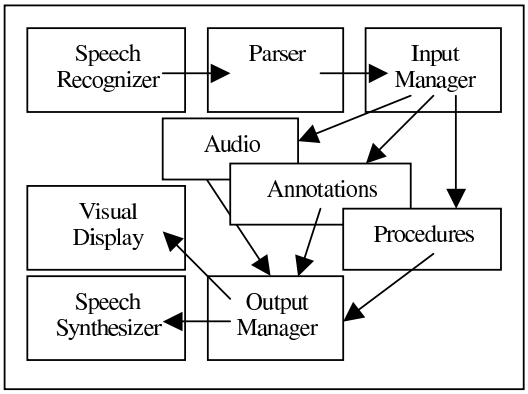

Figure 1. December 2002 Checklist architecture.

3.4 Dialogue Input Management

Each behavior agent has an agenda (Rudnicky and Xu 1999) of the types of input it is expecting. The behavior agents are maintained in a priority queue according to recency of use. Incoming interpretations such as "load(water)" or "increase(volume)" are matched against each behavior agent in turn. When a match is found, that behavior agent is promoted to the top of the queue and the message is dispatched to the agent. This scheme allows us to coordinate multiple behavior agents. Although in the December 2002 implementation the agenda is fixed for each dialogue agent, a better extension would make the agendas dynamic in response to changes in dialogue state.

[1] At one point we were labeling each behavior agent a "dialogue manager". This resulted in calling the input manager the "dialogue manager manager"; such reduplicative terminology seemed baroque, so we fixed it.

3.5 Dialogue Behavior Agents

The Checklist system is capable of a number of functions, as provided by the following dialogue behavior agents.

Procedure agent (RavenClaw – Bohus and Rudnicky 2002). Available functions include the following.

Loading a procedure by saying, for example, "Load water procedure". The procedure is loaded from disk as a XML document and converted into HTML via XSL, and then rendered using Cascading Style Sheets (CSS). At the same time, the procedure is processed using XSL into a task description for use by the task-oriented dialogue management component (RavenClaw).

Asking yes/no questions of the user, for example "Are you ready to begin the procedure?" when indicated by task constraints or by the structure of the procedure itself.

Navigating through the procedure one line at a time ("next line") or one numbered step at a time ("next step"), and returning to previous lines ("previous line") or previous numbered steps ("previous step").

Annotation agent handles a variety of tasks.

Requesting a list of available images by saying "What pictures do you have?".

Requesting a specific image by saying, for example, "Show me the small waste water bag."

Taking a voice note by saying, for example, "Take a voice note."

Audio agent handles request to change settings.

Setting an audio alarm by saying, for example, "Set an alarm for three minutes from now."

Controlling audio output volume by saying, for example, "Speak up", or "Quieter."

3.6 Dialogue Output Management

Output requests from dialogue agents enter a common output queue, where they are transformed into a series of one or more display actions and/or strings of text to speak. The resulting actions are played one at a time.

3.7 Visual display

The procedure is displayed as an HTML document (Figure 2), with the current step highlighted in light blue, and the current substep (if any) displayed in black text on a white background (to contrast with the light blue of the current (larger) step).

3.8 Speech synthesis

Speech synthesis is handled by AT&T's speech synthesizer, equipped with a customized pronunciation dictionary.

4 Particular challenges

Particular challenges to be addressed for the Checklist system include telling whether the user is speaking to the computer or to another person, also known as "open mic" (work completed and currently being evaluated), and robustness to the noisy environment on board the Space Station.

5 Example dialogue

[System] Welcome to the Intelligent Procedure Assistant.
[User] Load water sampling procedure.
[System] I am loading the procedure. Okay, I've loaded it. The full title of this procedure is … (the title). Are you ready to begin the procedure?
[User] Yes.
[System] I have a note at this point. Shall I read it?
[User] No.
[System] Okay, now moving to step 1. (reads step 1 out loud).

6 Demonstration scenario

Conference attendees will be able to use the system as an assistant while performing a (simulated) Space Station task, such as collecting and analyzing a water sample from the drinking water supply. Participants may also have an opportunity to annotate the procedure using voice notes, and use other features as time permits.

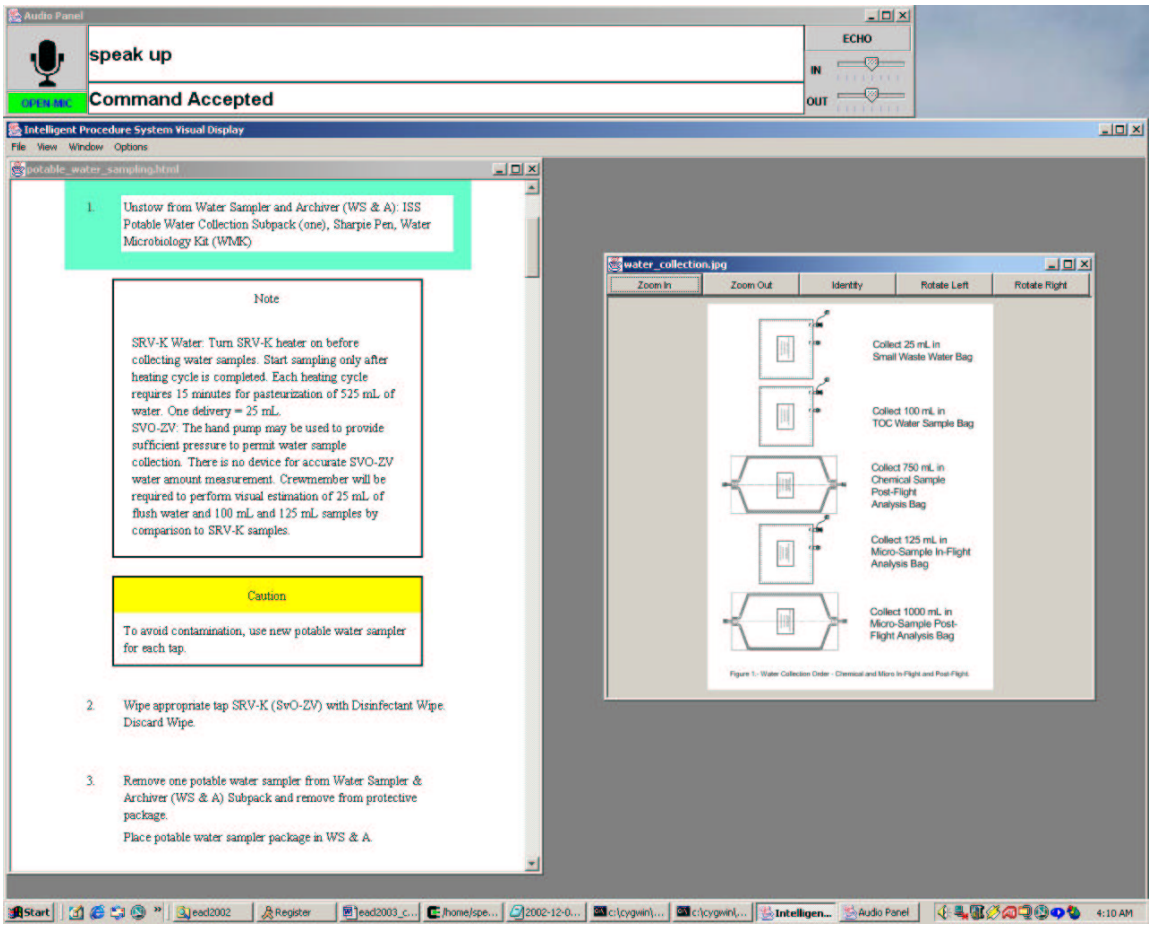

Figure 2. Visual display of December 2002 Checklist system.

References

Aist, G., Dowding, J., Hockey, B.A., Hieronymus, J. 2002. A Demonstration of a Spoken Dialogue Interface to an Intelligent Procedure Assistant for Astronaut Training and Support, ACL 2002, Demo Session, Philadelphia, July 7-12.

Allen, J., Ferguson, G., and Stent, A. 2001. An architecture for more realistic conversational systems. In *Proceedings of Intelligent User Interfaces 2001 (IUI-01)*, Santa Fe, NM, January 14-17, 2001.

Bohus, D., and Rudnicky, A. 2002. LARRI: A Language-Based Maintenance and Repair Assistant. IDS-2002, Kloster Irsee, Germany.

John Dowding, Jean Mark Gawron, Doug Appelt, John Bear, Lynn Cherny, Robert Moore, Douglas Moran. 1993. Gemini: A Natural Language System For Spoken-Language Understanding. Meeting of the Association for Computational Linguistics.

Rayner, M., Dowding, J., and Hockey, B. A. 2001. A baseline method for compiling typed unification grammars into context-free language models. Proceedings of Eurospeech 2001, Aalborg, Denmark, pp. 729-732.

Rayner, M., and Hockey, B. A. 2003. Transparent combination of rule-based and data-driven approaches in a speech understanding architecture. EACL 2003, Budapest, Hungary.

Rudnicky, A. and Xu W. 1999. An agenda-based dialog management architecture for spoken language systems. IEEE Automatic Speech Recognition and Understanding Workshop, 1999, p I-337.