N90-25522

An Expert System to Advise Astronauts During Experiments: The Protocol Manager Module

Guido Haymann-Haber*,
Silvano P. Colombano**,
Nicolas Groleau****,
Don Rosenthal**,
Peter Szolovits***,
Laurence R. Young****

* Knowledge Systems Laboratory, Stanford University, Palo Alto, CA 94304, U.S.A.
** Artificial Intelligence Research Branch, NASA-Ames Res. Ctr. Mail Stop 244-17, Moffett Field, CA 94035, U.S.A.
*** Laboratory for Computer Science, Massachusetts Institute of Technology. Cambridge, MA 02139, U.S.A.
**** Man-Vehicle Laboratory, Massachusetts Institute of Technology. Cambridge, MA 02139, U.S.A.

Abstract

Perhaps the scarcest resource for manned flight experiments - on Spacelab or on the Space Station Freedom - will continue to be crew time. To maximize the efficiency of the crew, and to make use of their abilities to work as scientist collaborators as well as equipment operators, normally requires more training in a wide variety of disciplines than is practical. The successful application of on-board expert systems, as envisioned by the "Principal Investigator in a Box" program, should alleviate the training bottleneck and provide the astronaut with the guidance and coaching needed to permit him or her to operate an experiment according to the desires and knowledge of the PI, despite changes in conditions. This report covers the Protocol Manager module of the system. The Protocol Manager receives experiment data that has been summarized and categorized by the other modules. The Protocol Manager acts on the data in real-time, by employing expert system techniques. Its recommendations are based on heuristics provided by the Principal Investigator in charge of the experiment. This prototype has been developed on a Macintosh II by employing CLIPS, a forward-chaining rule-based system, and HyperCard as an object-oriented user interface builder.

Introduction

There is an ongoing interest in understanding the phenomenon of space motion-sickness. This is a pervasive problem that, aside from producing discomfort to the astronauts, has at times significantly reduced their effectiveness during space missions.

In trying to shed light on the physiological principles that underlie this phenomenon, several experiments have been conducted both during space missions and on the ground. While the Principal Investigator (PI) in charge of the experiments can supervise the tests conducted on the ground, the astronauts have to act both as subjects and operators of the experiments while in space. Without the supervision of the PI, the data collected during past space trials has not been as complete and useful as desired. The crews have been unable to deal adequately with unexpected events, and make the necessary adjustments to the experiments. The PI, on the ground, cannot always be counted on for directions as he or she may have neither the accurate real-time data needed to make a decision, nor be able to communicate with the astronauts. The astronauts are generally unprepared to make these decisions, as the numerous activities they are expected to perform during the mission only allow them to have a basic idea of the nature and objectives of the experiments.

The experiments are conducted throughout the space mission during several sessions of approximately one hour. The schedule and content (the <u>Protocol</u>) of the sessions is prepared before the mission is launched. Once the mission is launched however, several events can affect that plan and necessitate changes to either the schedule or the Protocol of the sessions. These events include equipment malfunctions, running short of time, finding interesting data that the PI may want to investigate, or having a sick or otherwise unavailable astronaut.

We are proposing to provide a computer-based advice system that will help the astronauts to cope with these problems by assisting them in the detection of out-of-bounds conditions, in analyzing these conditions, and in suggesting alternative courses of action. With this system, the astronauts will have available some of the expertise of the Principal Investigator. The system is thus appropriately called *PI-in-a-Box*, or [PI] for short.

The next set of experiments will be conducted during the Space

Laboratory Mission (SLS-1) of the Space Shuttle, currently scheduled to be launched in 1990. A prototype of some of the modules of the system will be tested in flight in 1992 and on the ground during 1990.

The key success factors for the system are:

- The astronauts must perceive that the system provides obvious advantages for them.
- The use of the system must be optional. The experiment must be able to proceed independently of whether the system is being used or not.
- The implementation must be as hardware-independent as possible since the alternatives that eventually will be available during the space mission are unknown.

The subject of this report, is one module of *PI-in-a-Box*, the Protocol Manager. Reference 3 provides detailed descriptions of the design and implementation of this system. Early prototypes of other modules are described in reference 1.

Domain Description

This section provides an overall description of the domain addressed by this particular system. References 5 and 6 provide detailed discussions of the underlying physiological issues.

The experiment addressed by this system is called the Rotating Dome Experiment. Its purpose is to study the interaction of several spatial orientation senses during and following adaptation to weightlessness. Normally all the senses (visual, vestibular, proprioceptive, tactile) act in harmony during voluntary head movements. In orbit, however, the organs of the inner ear, no longer produce signals which the brain can use to deduce the angular orientation of the head with respect to the vertical - and of course the vertical itself ceases to have any real significance. Nevertheless, the brain still searches for a reference system, within which it can place external (seen) and body position measurements. Visual cues, both static and dynamic, as well as localized tactile cues, may become increasingly important in signalling spatial orientation.

A better understanding of the level of brain adaptation to altered gravio-inertial forces may help to explain and possibly alleviate the symptoms of space motion sickness, which are thought to be related to sensory-motor conflict concerning spatial orientation.

During the Dome Experiment, the subject's field of vision is filled by a dome. This dome rotates at various speeds and directions, while several measurements are being taken. The dome operation normally entails a one hour experiment with two astronauts - alternating as subject and operator. This period, referred to as an Experiment Session, is repeated several times throughout the space mission. In addition, the experiment is also performed on the ground during the days preceding the flight in order to get baseline data, and immediately following the mission in order to study the readaptation to the Earth's gravity.

An Experiment Session starts with un-stowing, setting-up and testing the equipment, and preparing the subjects.

The experiment is paced by a dedicated computer, the Experiment Control and Data Systems (ECDS), which generates instructions, starts and stops the dome rotation according to pre-programmed sequences, acquires, digitizes and transmits data, and permits routing of analog test signals for hardware testing and for calibration.

Each subject will normally take part in three runs under different conditions:

- The free float condition has the subject restrained by a bite-board and his or her right hand on a joy stick.
- The neck twist condition is like the previous one, except that the subject starts each dome trial by tilting his or her neck.
- The bungee condition has the subject held down to a foot restraining grid plate by stretched elastic bungee cords.

Each run lasts about 3 minutes. After the experiment, the equipment is deactivated and stored.

During the course of an experiment several types of data are recorded. These include:

A joy stick signal from a potentiometer adjusted by the subject. The subject uses it to indicate the strength of his or her visually induced rotation rate relative to the speed of the dome. Full deflection of the potentiometer clockwise, for example, would indicate that the subject felt that he or she was rotating to the right and that the dome (which was actually turning counter-clockwise) was apparently stationary in space.

A bite-board measures neck torque by means of strain gauges attached to the support. It measures the tendency of a subject to straighten out his or her head to the upright when sensing that he or she is falling.

Neck muscle EMG from the right and left sides are also indicators of the initiation of righting reflexes to straighten the head.

The astronauts perform the experiment by following a checklist with detailed step by step instructions. This checklist is prepared by the PI before the space mission. Unfortunately, the astronauts often must deviate from this pre-defined protocol due to a variety of circumstances such as:

- The experiment is running late. This could, among other things, be due to a late start or delays in performing some of the steps of the experiment. Since the ending time of the session is strictly enforced, some parts of the experiment may have to be eliminated.
- There are equipment problems. A piece of equipment may have failed, possibly degrading the quality of the data. A decision has to be made as to whether to continue the experiment with degraded data or to spend valuable session time trying to troubleshoot and fix the problem.

There are some additional circumstances in which a change in the protocol might be desirable, and that are very difficult for the astronauts to perceive, such as:

- The data being collected from the subject is "interesting." It might be desirable to perform some additional runs on that subject.
- The subject is providing "erratic" data that is not very useful. It might be desirable to concentrate on the other subject.

Communication channels between the spaceship and the ground may not be available for experiment use during a session. Consequently, the PI generally does not have real-time access to the data or the astronauts. As a matter of fact, even if this were possible, he may not have enough time to analyze the data and make a recommendation. In previous missions, where similar experiments were conducted, a significant amount of potentially useful data was never collected due to circumstances such as those mentioned above.

The PI-in-a-Box System

The *PI-in-a-Box* system has been divided into several relatively independent modules. It is centered around the Protocol Manager, which is the subject of this report. The Protocol Manager monitors and suggests proper actions during an experiment session. In addition, the system includes the following modules:

- A Data Quality Monitor, that monitors the output from the data collection system and pinpoints suspect signals.
- A Diagnosis and Troubleshooting System, that assists in the diagnosis and repair of equipment.
- An Interesting Data Filter, that detects interesting or unexpected patterns in the measurements.
- An Experiment Suggester, that comes up with additional tests that might be run if spare time is available.
- A Scheduler, that does long term scheduling of experiment sessions throughout the mission (not to be confused with the scheduler of an operating system).
- An Experiment Controller (ECDS), that controls the operation of the dome.
- A Signal Processing module that picks up the signal from the Experiment Controller (ECDS).

These modules interact with each other and with the Astronauts and the PI. A diagram of their interactions is shown in figure 1. It is important to note that the arrows represent the flow of control, not data. The latter may move directly between any two modules as needed.

The system must operate in real-time. It needs an Executive to schedule the execution of the appropriate module, since different events may compete for computing resources. The aim is to make the system as independent as possible from the hardware or operating system configurations on which it may be implemented. Consequently, few assumptions have been be made about the architecture of the Executive in the design of each of the modules, and standardized interfaces have been defined.

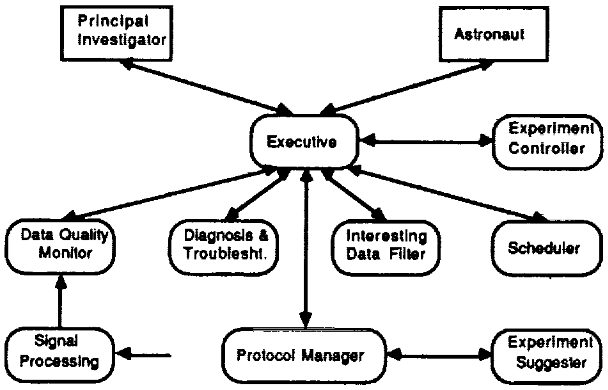

Fig. 1: Control flow between the modules

Because of the complex user interfaces required by *PI-in-a-Box*, the development of this project has been done on the Apple Macintosh II. Since the computers used during the mission have to be space-qualified, which entails a series of rather lengthy and expensive tests, there is no guarantee that a Macintosh will be available during the mission. Consequently, the system must be developed with the ability to be ported in relatively short order to some of the available hardware platforms.

The Protocol Manager uses a combination of a rule-based system in order to do the "reasoning," and a fast application builder in order to build the user interface.

For the "reasoning" part, CLIPS (*C Language Integrated Production System*) was used (ref. 2). CLIPS is a forward-chaining rule-based system that combines some of the features of the expert system shells of OPS5 and ART. CLIPS was developed and is supported by NASA. It was chosen because currently there are versions for both the Macintosh and 80n86-based computers, making it easy to port between any of the environments. In addition, the CLIPS source code, written in C, is available and can be customized in order to handle specific needs. Finally, CLIPS is a fairly simple and yet powerful language.

The user interface was built using HyperCard (ref. 4). Although it has several important restrictions, HyperCard provides a good environment to build complex user interfaces.

The general philosophy has been to shift as much as possible of the "reasoning" processes to CLIPS. CLIPS is easy to port to other environments, while HyperCard probably will be replaced in the final version.

Using the Protocol Manager

Most of the visible activity of the Protocol Manager is centered around the Session Manager. A typical screen is shown in figure 2.

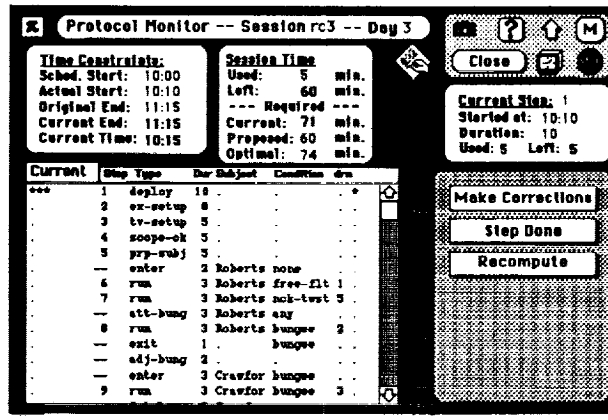

Fig. 2: Typical Protocol Manager screen

In this example, the astronaut has just started a new session. The top box indicates that this is the third day in the mission and that this session is code-named "rc3."

The Time Constraints box shows that this session was scheduled to start at 10:00. However, the astronaut has indicated that it actually started at 10:10. The scheduled ending time remains unchanged at 11:15. The current time is 10:15, that is, 5 minutes into the session.

The Session Time box indicates that 5 minutes of the current session have been used and that 60 are left. Below, the Session Manager reports that 71 minutes would be required in order to complete the protocol proposed by the PI. This is 11 minutes past the scheduled ending time. However, the Session Manager indicates that it has an alternative protocol to propose. It would fit within the time that is available, taking exactly 60 minutes. There also is an "optimal" protocol that includes everything that the Protocol Suggester would like to try. It takes 74 minutes.

The protocol steps themselves are displayed in the window below. The stars (***) identify the step that the Protocol Manager believes that is currently being performed. For each step there is an associated step number, a description (Type), an expected duration (in minutes), and if applicable, the name of the subject, the condition and a dome run number (drn). The astronauts for this session are Roberts and Crawford.

The Current Step box on the right-hand side indicates some basic data about the current step.

The system includes on-line help and the possibility to enter user comments. The protocol display may also be used as a checklist. By clicking on the appropriate step, the detailed instructions that need to be performed are displayed.

The astronaut may examine the alternative protocol that has been proposed by clicking on the **Current** heading, and selecting the **Proposed** protocol, as shown in figure 3.

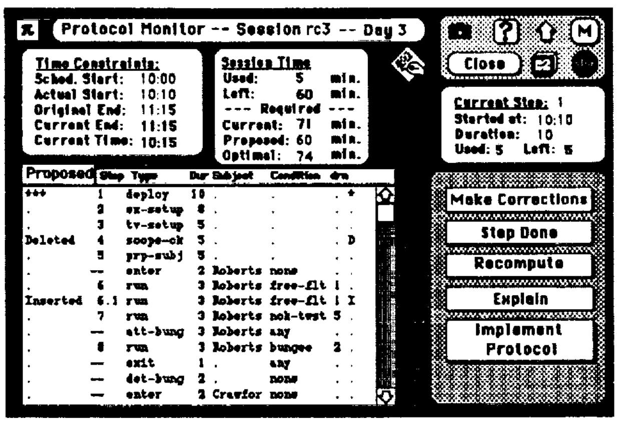

Fig 3: Alternative protocol proposed at the start of the Session

One of the changes that is being recommended is to delete the *scope check* step. In addition the Protocol Suggester recommends to insert a second *free float* run (step 6.1). The astronaut may click on step 6.1 in order to get an explanation for this recommendation, as shown in figure 4.

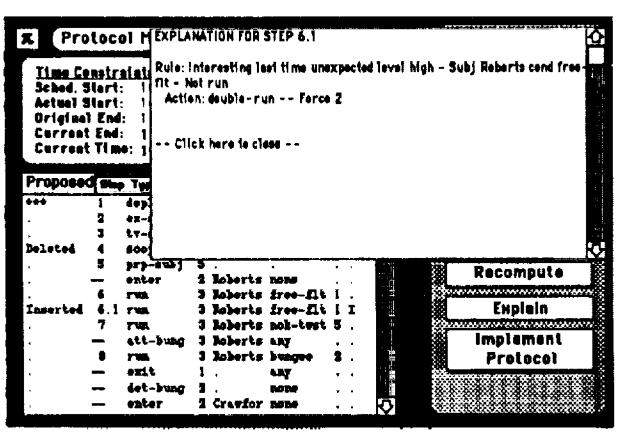

Fig. 4: Explanation for recommended step 6.1

The explanation indicates that interesting data was found on Roberts during the last session and was not investigated. This triggers a request for a double run of the *free float* condition, with an associated force of 2 (see next section).

If the astronaut decided to adopt the proposed protocol, he or she just needs to click on the **Implement Protocol** button.

The Protocol Manager is designed to operate with a minimum of input from the astronauts. It will make the proper assumptions about the state of the experiment session based on the signals it gets from the other modules. The astronauts may modify any of these assumptions.

For instance, in the session displayed above, Roberts is scheduled to be the first subject. If after completing the preparations for the experiment, the astronauts decide to reverse this order, they may indicate this to the Protocol Manager, which will suggest a modified protocol. The protocol, as accepted by the astronaut, is shown in figure 5.

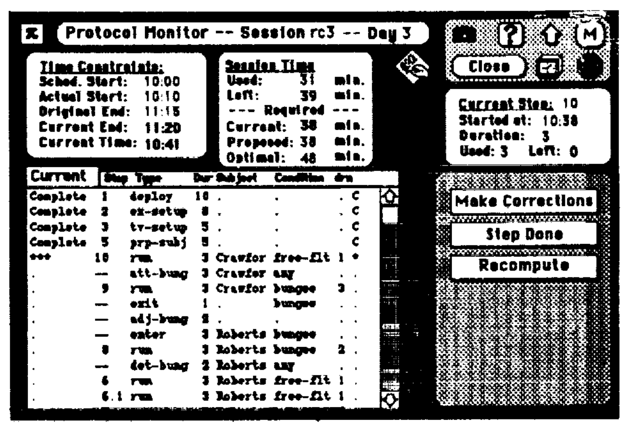

Fig. 5: Modified protocol with subjects switched

In the new protocol, the currently scheduled runs for Roberts are moved and re-inserted in a different order after the *bungee* run for Crawford. The run sequence for Roberts has been altered in order to take advantage of the fact that the bungee will be attached when Crawford exits the dome. This saves time.

Notice that a 5 minute time extension has been granted since the start of the experiment (the ending time is now 11:20). Nevertheless, step 11, the *neck twist* run for Crawford has been cut anyway in favor of a double *free float* run for Roberts. If the astronaut were to request an explanation, the contents of figure 6 would be shown.

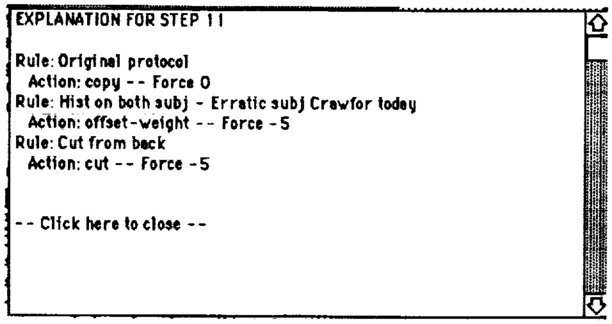

Fig. 6: Explanation for the elimination of step 11

Each step has an associated <u>weight</u>, which is the result of recommendations of varying <u>forces</u>. This concept is used in order to select the most important steps. It is explained in more detail in the next section.

The explanation depicted in figure 6 says that the *bungee* step was contained in the original protocol and is thus included with force 0. However, during the last run, Crawford's measurements were <u>erratic</u>, which prompted a reduction in the weight of his run by 5 units of force. Consequently, given that there was not enough time, the step was cut.

The Protocol Manager contains several other facilities that include the ability to study the history of previous sessions, undo actions, modify a series of decision parameters, and so forth.

Protocol Manager Architecture

The diagram of figure 7 presents an overall view of the environment in which the Protocol Manager operates:

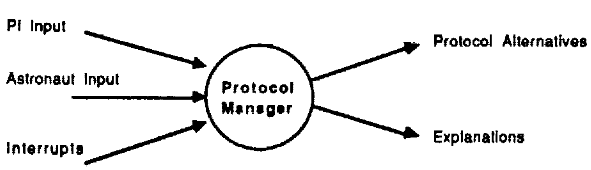

Fig. 7: Overall environment of the Protocol Manager

The PI and the Astronauts can interact with the Protocol Manager by entering, updating or requesting information. In addition, the Protocol Manager reacts to certain messages sent by the other modules of the system. These messages are referred to as <u>interrupts</u>. The Protocol Manager, in turn, provides protocol alternatives and explanations for its recommendations to its users.

The Protocol Manager has been broken down into two components. A <u>Session Manager</u> that handles all the interactive work, and a <u>Protocol Suggester</u>, that operates invisibly underneath the Session Manager and provides it with new protocol alternatives upon request. The Session Manager and the Protocol Suggester interact by passing data to each other as illustrated in figure 8. The arrows represent the data flow between the two components.

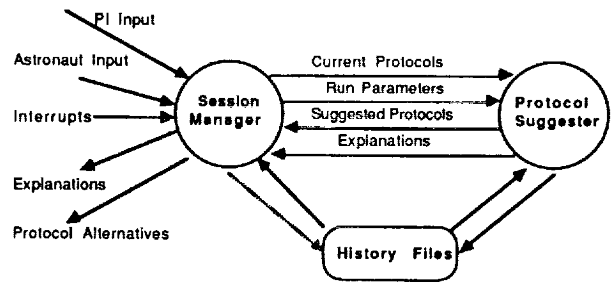

Fig. 8: Data flow between the Session Manager and the Protocol Suggester

The Session Manager periodically requests new protocol alternatives from the Protocol Suggester. As illustrated in the diagram, the modules communicate by sending data directly to each other, and through history files where all relevant events that have occurred during the mission are saved. Note the shaded link from the Session Manager to the History Files. This is a "development link" used during the testing phase of the system in order to set up different scenarios.

There are three main motivations that favor this division of the Protocol Manager:

- The Protocol Manager must be "aware" of time. It must be available upon request and it must take actions that are triggered at certain time intervals. Most rule-based systems (and CLIPS is no exception) operate on a run-cycle concept; they read input values, fire the appropriate rules, and produce a result. During this period, conditions are assumed to stay constant, that is, time is static. This is clearly not an adequate environment for the above system. By virtue of this division, the Session Manager acts as a supervisor that decides when it is necessary and possible to update the suggested protocol.

- The Protocol Manager must receive input from the other modules and the users in real time. It must provide reasonably fast responses to what in general are simple requests (such as a request to display the instructions for a particular step). For the reasons stated above, CLIPS is not appropriate for this type of operation.

- Most of the "intelligence" of the system lies in the process of suggesting a protocol, and a significant effort must be put into its development and maintenance. It is very desirable that the implementation of this process be as independent as possible from the hardware or system software. By making the Protocol Suggester a "black box" with a minimum of assumptions about the operating environment, the modifications that may result from system configuration changes are limited mostly to the Session Manager.

The **Session Manager** then, handles the interaction with the users and the interface with the other components of *PI-in-a-Box*. In essence, the Session Manager provides the astronauts with an intelligent checklist that displays the progress of the session and provides alternatives for action based on the output of the Protocol Suggester and its knowledge about the environment.

The **Protocol Suggester** is subordinate to the Session Manager, and provides it with new protocol alternatives upon request. In broad terms, the process of suggesting a protocol consists of three stages:

I Proposing a series of actions to take given the state of the current protocol and knowledge about the past history of the current and previous sessions.

II Generating all the steps that should be executed in order to comply with the proposed actions.

III Assembling the "best possible" protocol from those steps, that complies with the time constraints of the current session.

This process model represents a key decision in the design of the Protocol Suggester. During the conversations with the PI it became apparent that there were two sets of heuristics: heuristics to decide which steps to include in the protocol, and heuristics to decide in which sequence to perform the steps.

Since generally there are more steps that are desirable to perform than there is time to actually perform them, a complex interaction ensues between all the different heuristics in order to decide which particular steps to perform in any given context. There is clearly the potential for an explosive growth of the number of combinations. This could make the system unmanageable, un-maintainable, and slow.

The solution adopted was to introduce the concept of step <u>weight</u>. Each step has a weight associated with it. This weight reflects the importance of the step, the higher the weight, the more desirable it is to perform the step. Through this artifact, the problem is broken down into two independent parts: determining which steps to perform, and choosing and ordering the steps with the highest weights that fit within the allotted time. The former is performed by stages I and II, while the latter is done during stage III.

There may be one or more heuristics which favor the inclusion of a particular step. These heuristics are expressed in stage I by proposing actions. Each of these actions has an associated <u>force</u>. The forces of all the actions proposing a particular step are combined in order to produce the weight of that step. The current heuristic is to simply make the weight of the step equal to the highest force of all the actions that propose that step. This is done as part of stage II.

While the use of weights is a completely arbitrary solution, it has provided a surprising flexibility in adjusting the actions for each scenario. The main disadvantage of this approach is that in the explanations for the inclusion or exclusion of a step, the causal chain that leads to the result is somewhat blurred.

The main advantage of the "weight" approach is, of course, the avoidance of a combinatorial explosion of rules. Adding a new rule is mostly a linear process, with few, if any, side effects to the other rules. Another advantage is that the system is more robust; if a particular combination of circumstances has not been contemplated, the Protocol Suggester will provide a reasonable answer, even though it may not be the best.

After each invocation, the Protocol Suggester returns the following information to the Session Manager:

- An <u>optimal protocol</u>, that is, a protocol that includes all the steps that the Protocol Suggester would like to see executed, without regard to the time it would take to perform them. In other words, all the steps generated during stages I and II are included, regardless of their weight.

- A proposed protocol, that is, a protocol that fits within the time currently allotted to the session. This protocol is a subset of the optimum protocol. However, the steps may be in a different sequence.

- A set of explanations, justifying the inclusion or exclusion of each step from the protocol.

Conclusions

This prototype system has shown that it is possible to design a fairly sophisticated experiment protocol manager for use in space. Furthermore, it is possible to do it with relatively unsophisticated hardware and software.

The main challenge in the design of such a system is to conceive a suitable software platform that provides a good paradigm for future growth and maintenance of the system. We believe that this has been achieved.

Important challenges lay ahead. These include the ability to make all the modules work together in real-time, and producing a good user interface.

Acknowledgements

This project was made possible by NASA grant NCC 2-570 and RTOP 506-47-11 for "Crew Station Design," respectively from the AI Research Brand and the Human Factors Division at NASA-Ames.

The Stanford University Knowledge Systems Laboratory and Apple Corporation have provided generous support to this project.

We would also like to acknowledge the contribution of Dr. Irving Statler and the members of the Human Factors Division. Dr. Statler's ideas and encouragement for serious user interface work will hopefully come to fruition in a later version of *PI-in-a-Box*.

References

1. Colombano, S., Young, L.R., Wogrin, N., Rosenthal, D. "PI-in-a-Box: Intelligent Onboard Assistance for Spaceborne Experiments in Vestibular Physiology." *Proceedings of the IV Conference on AI in Space Applications*, November 1988.

2. *CLIPS Reference Manual*. Version 4.2. Artificial Intelligence Section, Lyndon B. Johnson Space Center, April 1988. Program and manual available from "CLIPS Users Help Desk/M30. Computer Sciences Corporation. 16511 Space Center Boulevard. Houston, TX 77058. (713) 280-2233.

3. Haymann-Haber, G. *PI in a Box: The Design of an "Expert" Protocol Manager*, Knowledge Systems Laboratory Report 89-55. Stanford University, 1989.

4. *Apple HyperCard Script Language Guide: The HyperTalk Language*. Addison-Wesley, 1988.

5. Young, L.R., Shelhamer, M., Modestino, S.A. "M.I.T./Canadian Vestibular Experiments on the Spacelab-1 Mission: 2. Visual Vestibular Interaction in Weightlessness," *Experimental Brain Research 64:* 299-307, 1986.

6. Young, L.R., Oman, C.M., Watt, D.G.D., Money, K.E., Lichtenberg, B.K., Kenyon, R.V., Arrott, A.P. "M.I.T./Canadian Vestibular Experiments on the Spacelab-1 Mission: 1. Sensory Adaptation to Weightlessness and Readaptation to One-g: an Overview," *Experimental Brain Research 64:* 291-298, 1986.