Research Institute for Advanced Computer Science
NASA Ames Research Center

IN-81
43045
P14

An Application for Multi-person Task Synchronization

Robert L. Brown
Dee Doyle

(NASA-CR-187712) AN APPLICATION FOR MULTI-PERSON TASK SYNCHRONIZATION (Research Inst. for Advanced Computer Science) 14 p N92-11918
CSCL 05A Unclas
G3/81 0043045

RIACS Technical Report 90.24

NASA Cooperative Agreement Number NCC2-387

An Application for Multi-person Task Synchronization

Robert L. Brown
Dee Doyle

JULY 1990

The Research Institute of Advanced Computer Science is operated by Universities Space Research Association, The American City Building, Suite 311, Columbia, MD 21044, (301) 730-2656.

The work reported on herein was supported by Cooperative Agreement Number NCC2-387 between the National Aeronautics and Space Administration and the Universities Space Research Association.

1. Introduction

This paper describes a project to investigate computer applications that will enable a group of people to synchronize their actions when following a pre-defined task sequence. We assume that the people involved only have computer workstations available to them for communication. Hence, our approach it to study how the computer can be used to help a group remain synchronized.

It is our purpose to design and develop a series of applications that we can use as vehicles for experimentation. The series will incorporate increasingly more powerful capabilities, building on what we learn from previous versions.

An example of how this technique can be used for a remote coaching capability is explained in a report describing an experiment that simulated a Life Sciences experiment onboard Space Station Freedom, with a ground-based Principal Investigator providing the expertise by coaching the on-orbit Mission Specialist. For more information, see [Haines89a].

2. Background

When a group of people work together in tight collaboration on a pre-defined task, such as repairing a complicated device, performing a laboratory experiment, or preparing an aircraft for flight, the task can usually be described as a partial ordered graph of subtasks, where each subtask is an indivisible unit of work typically performed by a single individual. Figure 1 shows such a partial ordering. The graph is typically represented as a written set of instructions, as in a checklist or repair manual, or as a chart.

When the group of people are working in close physical proximity, synchronization is typically simplified by the use of verbal communication, or a task supervisor overseeing the progress. In the former case, the task graph can be shared, or replicated for each member. In the latter, typically only the supervisor has the task graph and gives instructions or orders to each subordinate member. However, when the group is geographically dispersed, such tight communication or supervision is not as simple. Radio links can be used for verbal and video communication, and the task graph is represented on paper and available to each member of the group. Voice and video can be used in the supervisor model, as well, with the supervisor parcelling out instructions in the correct order.

Though audio networks are commonplace (the telephone system), personal video networks are not commonly available. Digital computer networks, however, have become more and more available to the science community, and predictions are that the trend will continue for many years. Hence, we are investigating how these networks, and the workstations people use to interface to them, can be used to support collaborative task sequencing.

Computer workstations have demonstrated themselves as useful in a variety of collaborative tasks. Computer electronic mail can be used as a collaboration technique. A group of people working together can stay in close contact with each other and exchange documents. The granularity of interaction is too large, however, for electronic mail to be used for tasks whose subtasks are much finer grained than the exchanges.

One area of past work that is particularly relevant to this work is in multimedia conferenceing. *[continue on with MMCONF, Diamond, SLATE discussion].*

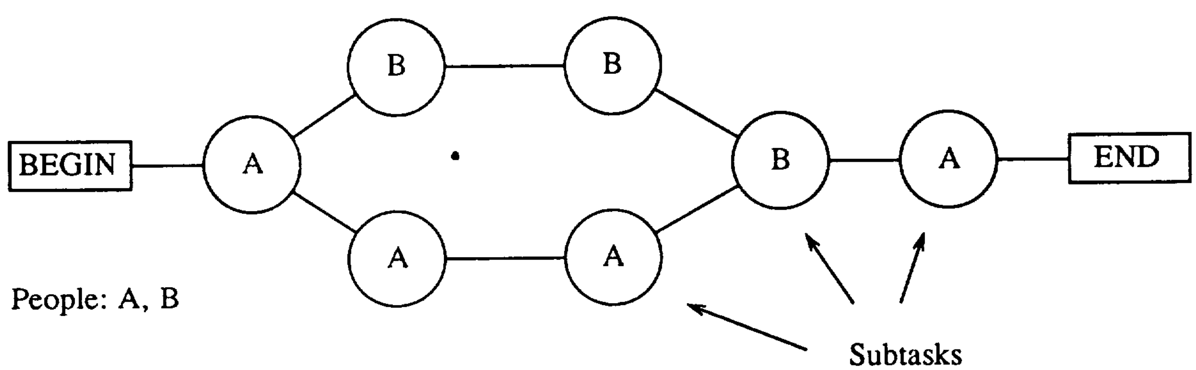


Figure 1. Partial Ordering of Subtasks

3. Possbilities for Computer-supported Task Sequencing

The remote coaching facility uses a simple sequencer in which the subtasks form a total ordering. Such task graphs can be called a "check list" because each subtask must be competed ("checked off") before then next one begins. There is no possibility for parallelism[1].

Complete seriality is not inherent to task sequencing; procedures often provide for simultaneous activities. A computer-support task synchronizer that supports simulteneous activities can also support completely serial activities by simply providing it with a serial specification. However, we do not yet understand all the issues related to task synchronization supporting simultaneous activities, or if such has practical application for dispersed collaborators. Hence, we chose to study only the completely serial case at first.

4. Design of the Project

We chose to approach this project incrementally. Because of the experiences gained in the Haines remote coaching study, we chose to mimic the software developed therein as our starting point. However, the Haines study software is developed solely for Apple Macintoshes and is written using the HyperCard application. Our computing environment relies on open systems, and our software platform includes UNIX, the X Window System, and the Motif toolkit. We did not restrict ourselves to any particular hardware platform.

We call the experimental steps in our project "phases," where each phase consists of a design of a tool and its implementation. Each successive phase will add functionality based on what we learned from the previous phase. The first phase is called "phase zero" because it directly emulates the software from the remote coaching facility and does not provide any new functionality. We expect a high degree of code sharing between phases, but would not hesitate to begin anew at any step.

[1] This does not strictly apply to the remote coaching software, which contained the possibility of a subtask requiring the user to gather several items in any order from a locker.

5. Phase Zero

As metioned, the phase zero software we designed to closely mimick the behavior of the checklist application in the remote coaching application. However, we sought to develop the application using open systems software technology and hence, using the X Window System from Project Athena at MIT. We succeeded in this goal and have a configurable distributed checklist application that is transportable across many workstation platforms. We have demonstrated this code by creating working displays on Sun Microsystems workstations, a Stardent Titan-1000, an Apple Macintosh II, and a NCD X-station, all at the same time.

5.1. Design

The design goal of phase zero was to allow the specification of the checklist task as a simple text file, that can be created using any standard text editor. We designed and implemented a straightforward textual language to describe the checklist. A grammar in BNF notation is presented in Figure 1. The application itself is composed of two major phases: input processing and interpretation.

The first phase is solely responsible for converting an external representation of the checklist into an internal representation, and building the tables describing the location of each partcipant at the same time. Because the participants can move about, the specification of their names and locations can only be known at runtime. Hence, that information is encoded as a part of the checklist specification. Arguably, the name & location information should be separated from the checklist description itself, but standard UNIX tools allow for this separation, and subsequent combination before the invocation of the checklist application.

5.2. Implementation

The distributed checklist, or DCL, application is constructed as a single process that simultaneously manages several displays, one per participant. This capability takes advantage of the ability of the X Window System to support remote displays, in fact, the DCL application code need not run on the same display workstation as any of the displays.

Critical to usefulness of a distributed checklist is keeping all participants aware of the state of the procedure. Hence, much of the activity of the application concerns assuring that all displays reflect the same state.

Figures 2-8 show various screen images from the application. The procedure displayed is one taken from the remote coaching experiment.

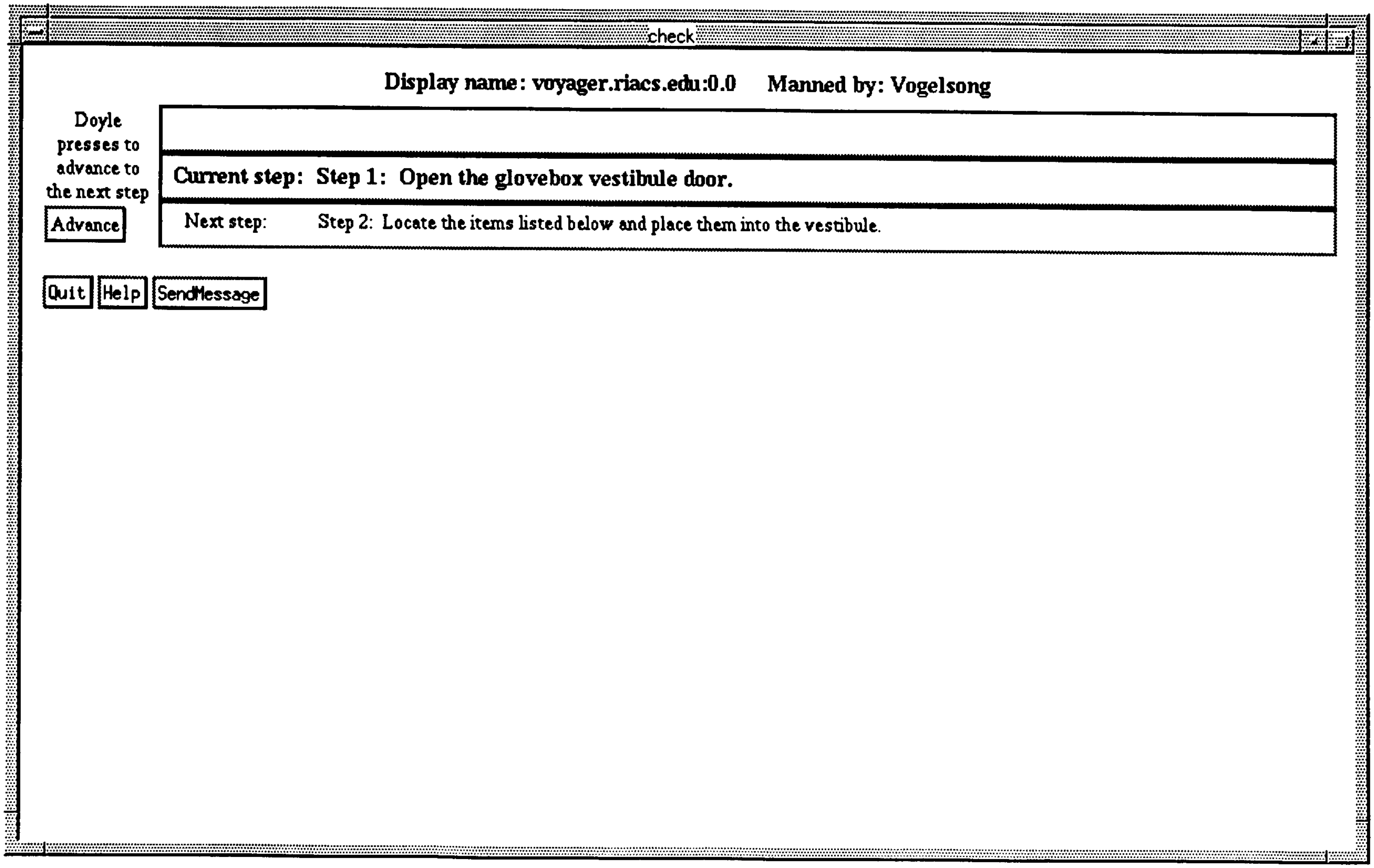
6. Conclusions and Future Work

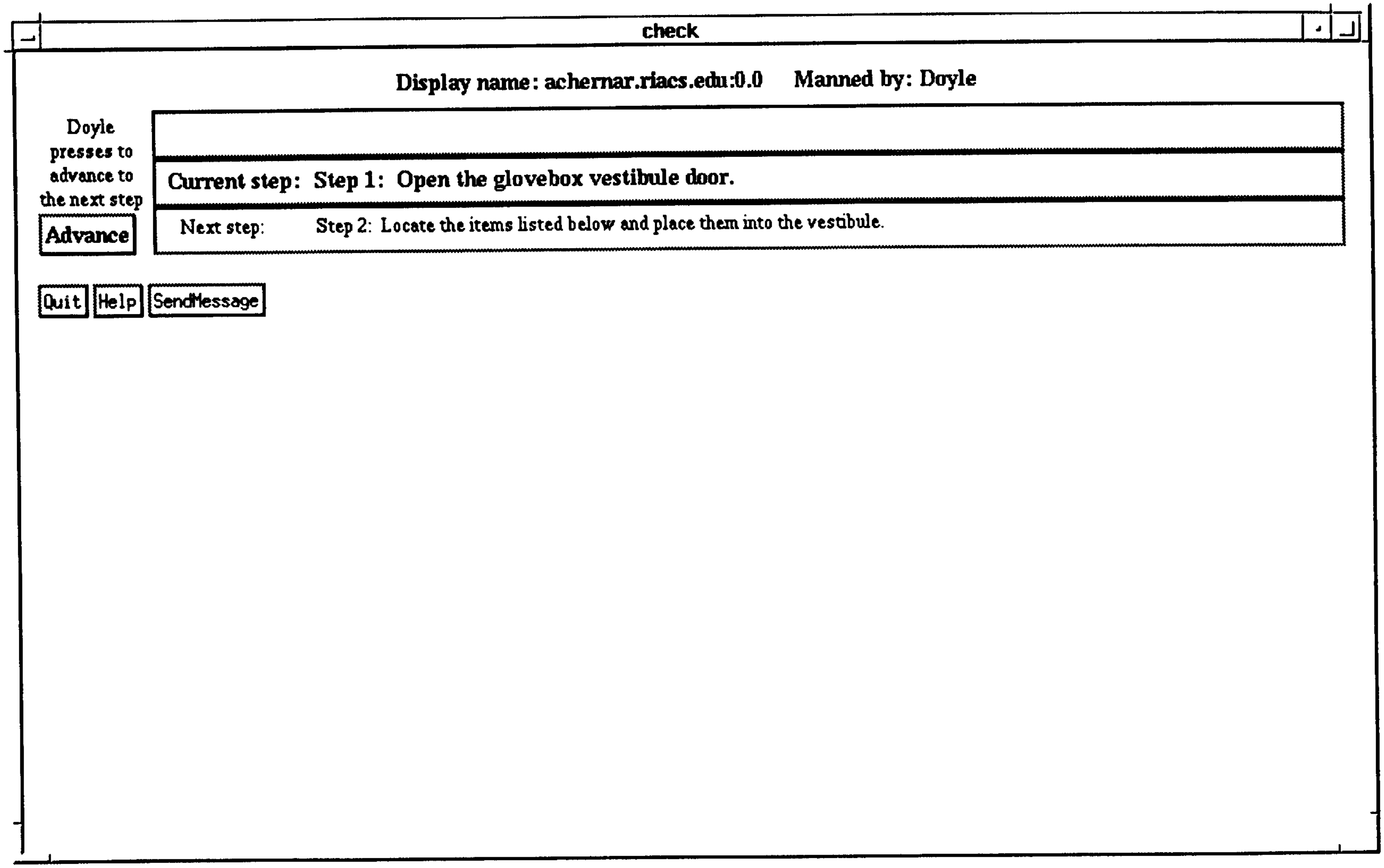
The usefulness of the distributed checklist has been demonstrated by its application in a remote coaching experiment. The suitability of the X Window System for this type of application is demonstrated by the ease with which the application was built. Without the use of a windowing system based on a server/client model, communication among the software driving the individual display would have to be explicitly coded into the application. This was the case the the HyperCard-based implementation on the Apple Macintosh systems. By allowing a single host to conrol the checklist, synchronizing the displays was simplified.

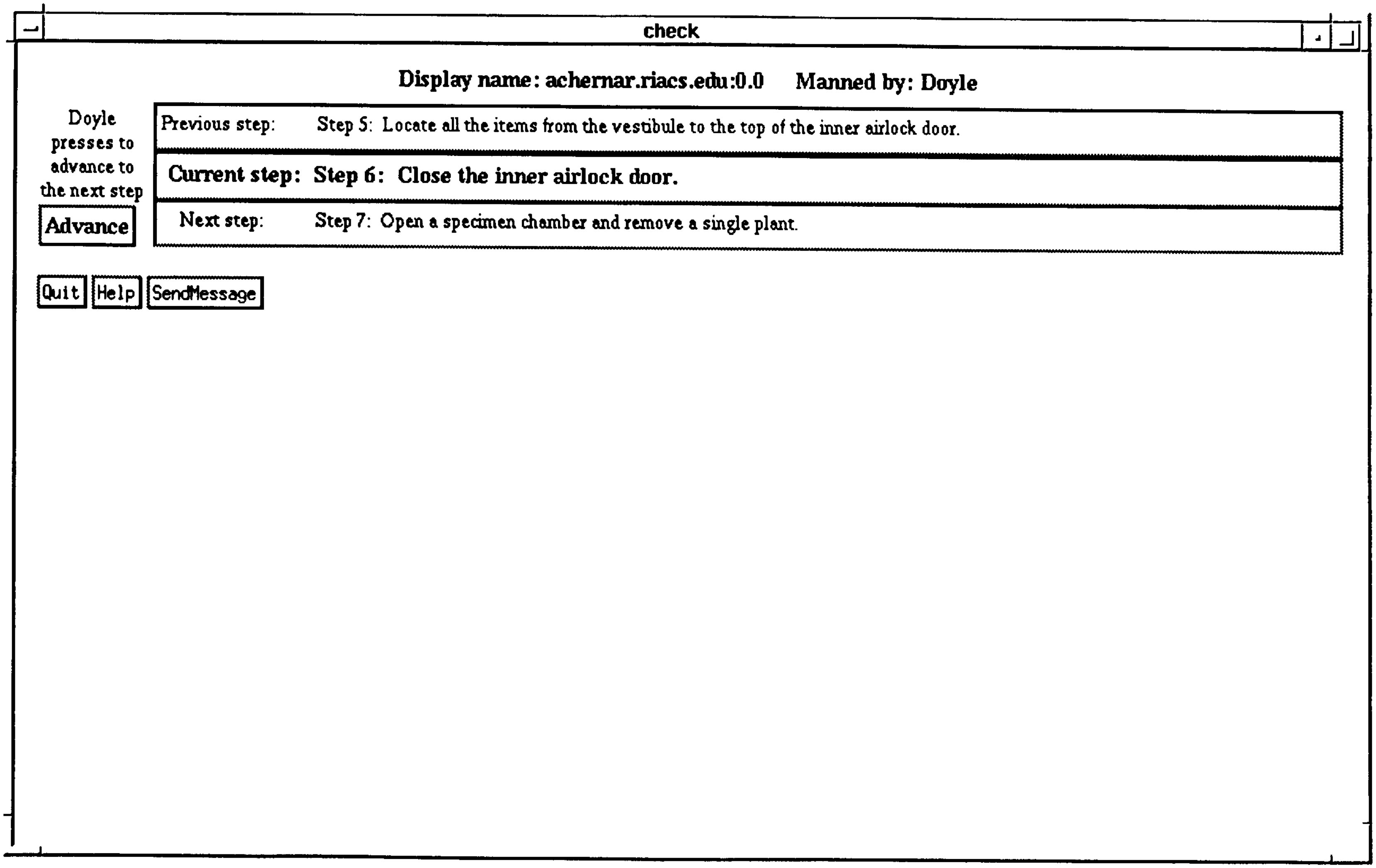
In the next generation of task synchronization application, if we pursue this project, will enable separate activities to be performed in parallel by different participants. We do not, however, have a target application for such a tool and will have to creat one in order to validate the approach.

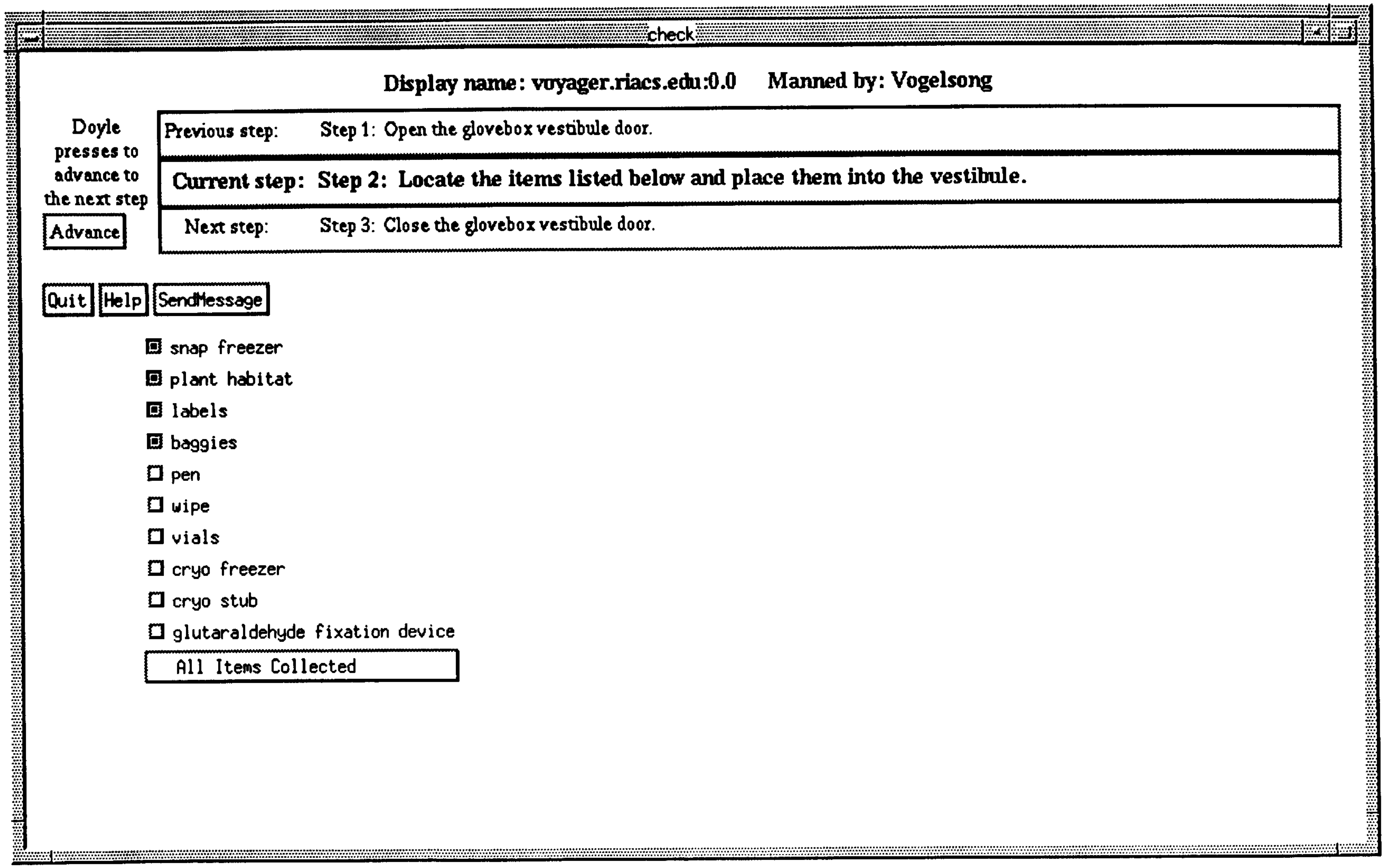
7. Appendix

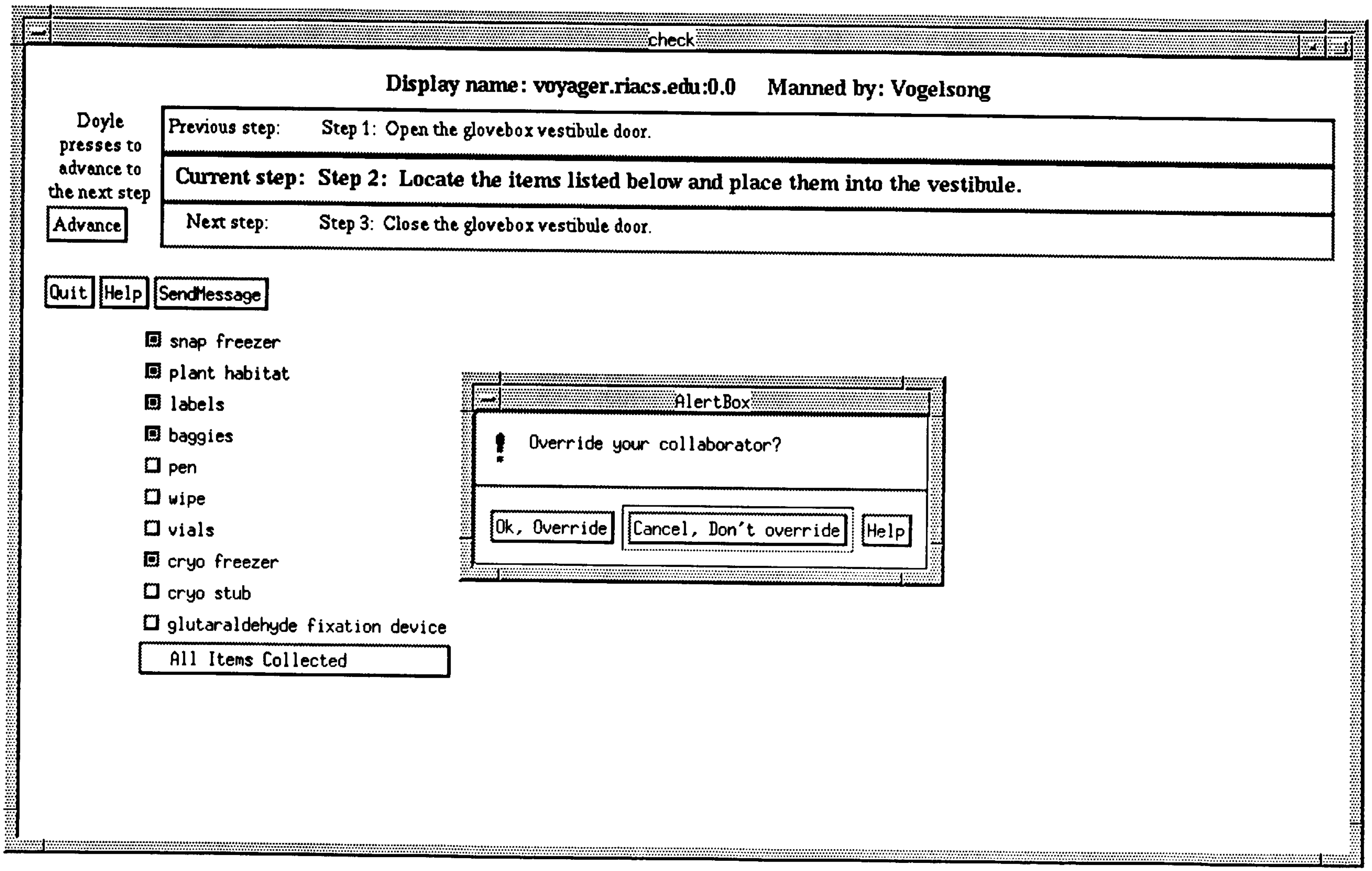
The following five pages show screen images from the distributed checklist application. The first page shows the initial state: Step 1 is in the middle of the list of steps and "Doyle" is the person listed as the authority for advancing the state. The second image shows the state of the screen for the other participant, "Vogelsong." The third screen shows the checklist of items that must be collected into the glovebox vestibule. The items on the list can be checked off in any order. The fourth screen shows what happens when someone other than the authoratative person selects to advance the list: a dialog box asks for confirmation. The final screen shows the checklist application in one of the latest stages; the previous instruction is displayed as well and the current and next instruction.


check
Display name: voyager.riacs.edu:0.0 Manned by: Vogelsong
Doyle presses to advance to the next step
Advance
Current step: Step 1: Open the glovebox vestibule door.
Next step: Step 2: Locate the items listed below and place them into the vestibule.
Quit
Help
SendMessage


check
Display name: achernar.riacs.edu:0.0
Manned by: Doyle
Doyle presses to advance to the next step
Advance
Current step: Step 1: Open the glovebox vestibule door.
Next step:
Step 2: Locate the items listed below and place them into the vestibule.
Quit
Help
SendMessage


check
Display name: achernar.riacs.edu:0.0 Manned by: Doyle
Doyle presses to advance to the next step
Advance
Previous step: Step 5: Locate all the items from the vestibule to the top of the inner airlock door.
Current step: Step 6: Close the inner airlock door.
Next step: Step 7: Open a specimen chamber and remove a single plant.
Quit
Help
SendMessage


check
Display name: voyager.riacs.edu:0.0 Manned by: Vogelsong
Doyle presses to advance to the next step
Advance
Previous step: Step 1: Open the glovebox vestibule door.
Current step: Step 2: Locate the items listed below and place them into the vestibule.
Next step: Step 3: Close the glovebox vestibule door.
Quit
Help
SendMessage
snap freezer
plant habitat
labels
baggies
pen
wipe
vials
cryo freezer
cryo stub
glutaraldehyde fixation device
All Items Collected


check
Display name: voyager.riacs.edu:0.0 Manned by: Vogelsong
Doyle presses to advance to the next step
Advance
Previous step: Step 1: Open the glovebox vestibule door.
Current step: Step 2: Locate the items listed below and place them into the vestibule.
Next step: Step 3: Close the glovebox vestibule door.
Quit
Help
SendMessage
snap freezer
plant habitat
labels
baggies
pen
wipe
vials
cryo freezer
cryo stub
glutaraldehyde fixation device
All Items Collected
AlertBox
Override your collaborator?
Ok, Override
Cancel, Don't override
Help