# TELESCIENCE: REMOTE INTERACTION WITH SCIENTIFIC EXPERIMENTS *

Larry C. Schooley and François E. Cellier
Dept. of Electrical and Computer Engineering
University of Arizona
Tucson, Arizona 85721

In this paper, two remotely controlled scientific experiments are described: remote control on an astrometric telescope, and remote control of a fluid handling laboratory operated by a laboratory robot under microgravity conditions. The aim of these seemingly quite different experiments was to study the feasibility of employing common generic technology for teleoperating scientific equipment in space.

## INTRODUCTION

NASA's Office of Space Science and Applications (OSSA) initiated the Telescience program to validate the user–oriented rapid–prototyping testbed approach to address a range of operations and information system issues. Fifteen universities, under subcontract to NASA, conducted a variety of scientific experiments emulative of the scientific research of the Space Station Freedom (SSF) era, and aimed at resolving critical issues in space station operations concepts and information system design. The goal was to allow scientists to interact with potential space station technologies in a manner that will allow resolution of design and specification questions without having to wait until space station hardware is available.

Telescience research can be split into three main areas:

(1) Remote control, sensing, manipulation, etc. This branch of Telescience is commonly referred to as *Teleoperations.*

(2) Remote data storage, searching, retrieval, etc. This aspect of Telescience is called *Teleanalysis.*

(3) Remote distributed management of scientific experiments. This part of Telescience is being referred to as *Teledesign* or *Telemanagement.*

The goals of our two telescience testbeds are related to the first area. The first testbed involves teleoperation of a forerunner of the Astrometric Telescope Facility (ATF), a proposed attached payload of the SSF. The second involves development of systems and software for Remote Fluid Handling (RFH) in support of the microgravity and life sciences. These seemingly quite different testbed demonstrations were selected to pursue the following goals in our research:

(1) To design a set of tools which allow teleoperation of scientific experiments. These tools include the man/machine interface at the remote commanding computer, the machine/machine interface between the remote commanding computer and the local controlling computer, and the machine/instrument interface between the local controlling computer and the equipment (robots, telescopes, measurement instruments, analytical instruments, etc.) which may comprise any specific experiment.

* Supported by the Research Institute for Advanced Computer Science (RIACS) of the Universities Space Research Association (USRA) under NASA subcontract 800-62

(2) To ensure that these tools are *generic* and *modular*, so that they can easily be applied to a wide variety of scientific applications on Space Station (or other platforms) and so that the individual modules can be revised without significant impact on the remaining parts of the software.

(3) To evaluate the technologies underlying the above developments and develop recommendations (specifications) for future development.

The next two sections of this report describe the results of the two scientific experiments, and the third documents the thread of common technology which underlies both testbeds.

## TELEOPERATION OF AN ASTROMETRIC TELESCOPE

Attached to Space Station, there will be installed an Astrometric Telescope Facility (ATF). Its primary objective will be the detection of planetary systems around other stars within $10 parsec$ of sol. It will be remotely operated from the ground. This experiment is a preliminary investigation of some of the issues involved in teleoperation of the ATF.

### Experiment Summary

In the first phase of this telescience project, a forerunner of the ATF (Thaw telescope, Allegheny Observatory, University of Pittsburgh) was simulated on a $\mu$VAX–II/GPX workstation. This telescope model was remotely operated from a second $\mu$VAX workstation. An observation scenario was developed on the host workstation using the Operations and Science Instrument System (OASIS) software package developed at the University of Colorado (Hansen, 1987). Commands are sent from the host workstation to the telescope workstation. Various real–time activities of the emulated telescope and data are telemetered back to the host workstation from the emulated telescope for display. This testbed is shown in Fig.1.

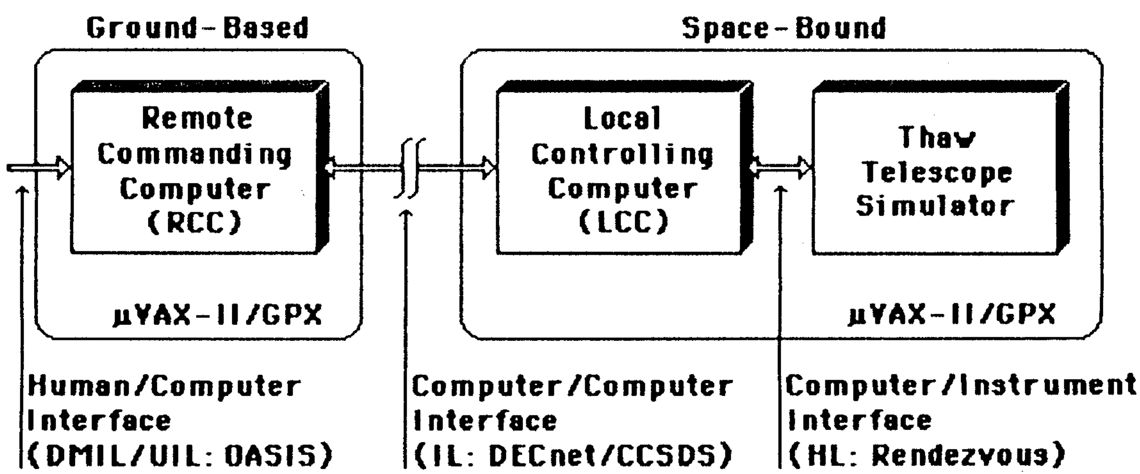


**Figure 1.** Astrometric Telescope Facility Testbed

The aim of this research was to analyze human/computer communication interfaces that are convenient for scientists who will remotely operate their instruments, as well as the computer system architecture and communications infrastructure which support these interfaces.

## Method of Investigation

Collaborators on this experiment included the Lunar and Planetary Laboratory of the University of Arizona, the Allegheny Observatory of the University of Pittsburgh, and NASA Ames Research Center. In several meetings, a preliminary set of parameters for the scenario of the astrometric telescope demonstration was selected. This led to a functional system design including a preliminary design of the intermediate language to be used between the local and the remote workstation (Cellier, 1987).

The simulation of the Thaw Telescope has been written in Ada, and emulates the functional operation of the telescope in the Observatory. Simulated telemetry and scientific data are sent back to the host workstation for processing and display. The observation scenario residing in the remote commanding computer is an OASIS application tailored to such purpose. In its initial phase, the teleoperation closely reflects the current way of operating the Thaw telescope. However, it is a subsequent goal of this project to experiment with alternative user interfaces in order to determine an optimally convenient way for the scientist to remotely interact with the Space Station ATF. The design of the improved human/computer interface will be done in close interaction with the scientists.

The remote commanding computer (RCC), a $\mu$Vax–II/GPX workstation, communicates at 9600*baud* over dialup phone lines, a Sytek network, or an Ethernet network with the local controlling computer (LCC) which, in our testbed, is represented by a second $\mu$Vax workstation. Intermediate command language statements, simulated telemetry, and simulated scientific data are exchanged using DECnet and CCSDS packets as the communication protocols. The second workstation includes the functions of the LCC as well as the telescope simulation.

The telescope simulation includes a failure mode simulator (Lew, 1988). Failure simulation is used for training purposes and also to aid in the design of a more robust remote control software. The failure simulation is accomplished by drawing a random number (uniformly distributed between 0 and 1) prior to command execution. This random number is then compared to a preset threshold which is user specified when the simulator is started. Whenever the random number is above the threshold, a failure in the command execution occurs, and the operator must then deal with this situation.

The communications functional design (Bienz and Hunter, 1988) contains provisions for future addition of multiple remote experimenters (multiple RCC's) connected through a communication network who will jointly control one experiment (one LCC). The software design will allow one OASIS user, holding an allocatable privilege key, to send intrusive telecommands, such as *slew the telescope*, or nonintrusive telecommands, such as *show telescope position*, to the simulator. Other (non–privileged) OASIS users may either issue nonintrusive commands to the LCC in order to observe the ongoing experiment, or may request a shadow image of the privileged user's session to be displayed. Non–key holders may also exchange messages with the privileged user and with each other (e.g., to request a transfer of the privilege key), and they can schedule a privileged session for themselves at some time in the future.

## Observations

(1) Remote operation of ground based telescopes is NOT commonly done, not because the control task is particularly difficult to solve (it is not), but because most telescopes are not properly equipped with the hardware required for remote operation.

(2) The most serious obstacle to remote operation of a telescope seems to be safety considerations. Most Earth–bound telescopes have not been equipped with sufficient safeguards that would allow for robust, error free, and accident free teleoperation of the instrument. Such problem can be avoided with properly designed hardware and software, however, it is much simpler and cheaper if an instrument is designed for teleoperation from the beginning, than to retrofit an existing instrument.

(3) There is currently no ground based astrometric telescope which is remotely controlled. Most instruments with some sort of remote operation facility are infrared telescopes and the rest are millimeter wave telescopes. In both these cases, the problems are quite different from those that can be expected when operating the ATF.

(4) Astronomers are not usually interested in data reduction techniques. For understandable reasons, they prefer to store every bit of information available to them. Consequently, little research has been done on data reduction for transmission or storage of star field images. However, the automatic control circuitry needs the star field information also (for the automated finder scope and guider scope control). The control circuits very definitely do not need the total star field information. On the contrary, the task of these circuits can be greatly simplified if the information contained in the star field is reduced to only those pieces that are needed for control purposes.

### Achievements

(1) Our development of the Thaw simulator has resulted in the definition of a set of sensors, controls, and safeguards needed for teleoperation of the Thaw telescope. Whether or not this is ever actually implemented, it has shown that the control, sensory, and safeguard systems of the ATF should be developed with the end user in mind. Successful telescience applications of this instrument call for a set of controls, sensors, and safeguards that should be defined early on, before the instrument has been totally developed.

(2) The goal of developing generic, modular software has been attained. Further discussion of this is contained in the RFH and TECHNOLOGY sections of this report. This result could not have been achieved in such a short time without the previous OASIS development by the University of Colorado.

(3) The work on video data compression for transmission of telescope pointing data has been completed (Walker, 1988). A form of run length coding, followed by variable word length (Huffman) coding was found to be the best for this application. This compression algorithm provides reduction from 8*bits* per pixel to 0.015*bits* per pixel for telescope pointing data. It appears that use of this technique will also allow compression of typical scientific observation data (of a general starfield) for storage or transmission by at least a factor of 10, and perhaps as much as 100.

## TELEOPERATION OF A REMOTE FLUID HANDLING LABORATORY

This experiment investigated the teleoperation of a robot controlled laboratory, such as those which will be installed on Space Station for scientific investigations in the Life Science and Microgravity Science disciplines.

### Experiment Summary

The initial demonstration involved teleoperation of a laboratory which provides automated handling and analysis of fluids, such as those which might be extracted from laboratory animals or human subjects as part of the life sciences program or sent from earth to be processed as part of the microgravity sciences program. The experiments selected were determination of the pH of a solution, and the separation of a solution into its charged components using electrophoresis. The laboratory equipment controlled included a Scorbot robot, special fluid handling apparatus, and the instruments and electronics required for the two experiments. The architecture of the teleoperations system is similar to that used for the ATF experiment previously described (see Fig.2).

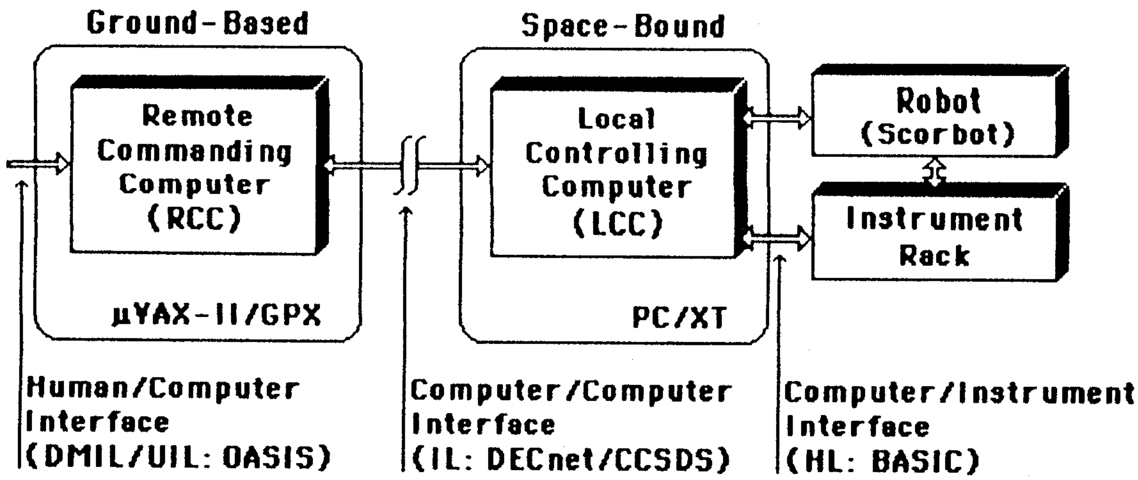


**Figure 2.** Remote Fluid Handling Laboratory Testbed

The aim of this research was to analyze human/computer interfaces which are convenient for scientists to remotely operate their instruments, as well as the computer system architecture and communications infrastructure which supports these interfaces. It was also intended to investigate the special hardware required for remote fluid handling under microgravity conditions.

## Method of Investigation

Collaborators on this experiment included the Center for Separation Science (CSS) of the University of Arizona, a predominantly NASA funded research facility, and NASA Ames Research Center. The experiments conducted for the initial demonstrations included electrophoresis testing and pH measurement. These experiments were performed by a remotely commanded but locally controlled laboratory robot. While the local control loop is fully automated, the telescience user can both monitor and direct his experiment through the remote command loop. Prototypes for the required circuitry for the teleoperation of the electrophoresis instrument and the pH instrument were designed and constructed, and the prototype setup was mounted on two mobile instrument racks in a configuration suitable for access by the robot.

A specialized syringe adapter that can be handled by the robot gripper was also designed and constructed (Raize, 1988). It consists of a cylindrical syringe–holder and a multi–purpose fixture for positioning the syringe. This device provides accurate positioning of all syringes for robot pickup, injection of precise fluid volumes, and disposal of used syringes. As an alternate to this syringe, which is activated by the robot gripper, a syringe driver assembly was designed with actuation by a stepper motor (Raize, 1988).

The actual fluid handling device was then converted to a motorized pipette. It was hoped that the use of this device would eliminate the need for the syringe holder and fixture, and reduce the overall weight of the components used in the system. Since the acquired pipette was designed for use in ground–based laboratory work and picks up fluids by creating a vacuum in a chamber above the disposable tip, changes were necessary to prevent the fluid from floating up this chamber in reduced gravity. This was done by using syringes with needles rather than the provided plastic tips, where each syringe was fitted with a plastic membrane attached to a spring, such that fluid pressure pushes the membrane/spring pair up in the syringe barrel, with the motorized pipette providing the driving force.

Present capabilities of the pipette permit four modes of operation at two speeds, namely pickup, dispense, multi–pickup, and multi–dispense at regular and fast speeds. The software which performed these operations in the original pipette (executed on a microprocessor placed inside the pipette itself) was replicated in Basic in the LCC (a PC clone), and the manual triggering which previously started each mode of operation (by pressing buttons located at the rear end of the pipette), was replaced by an appropriate command issued through the LCC. The original pipette motor control board was also replaced by special circuitry that provided the interface to the computer.

Holding of the pipette by the Scorbot gripper was impractical since the 12 inch length of the pipette and the attached syringe caused unacceptable positioning errors of the needle tip. Therefore, the entire wrist assembly was replaced by another assembly that holds the pipette. This assembly retains the up/down (pitch) motion of the original wrist and eliminates the no longer needed roll motion.

The remote commanding computer is the same $\mu$Vax–II/GPX workstation that has been used for the ATF experiment. The LCC is an IBM PC compatible with $640K$ memory, an 8087 coprocessor, a $20Mbyte$ hard disc, and a LabTender multifunction board.

The software interface design (Hack, 1988; Pan and Lew, 1988) consists of three parts: a human/computer high level macrocommand interface (an OASIS application) which allows the user to easily operate the laboratory from a remote location without having to be a fluent programmer, a machine/machine medium level command interface (intermediate language) which contains the set of commands internal to the system and enables the communication between the RCC and the LCC, and a machine/instrument low level command interface to the laboratory robot and the instrument rack. Depending on the type of robot used, the robot instruction commands may be further decomposed into a set of lowest level microcommands by a microprocessor located inside the robot itself. This was however not the case with the Scorbot used in this demonstration. In this design, high level commands are successively decomposed into a series of lower level commands which finally map into the hardware interface language of the used equipment.

Software modularity was a primary design goal. If the laboratory robot is to be replaced by another type of robot, only the code that maps the intermediate language interface into the machine/instrument interface needs to be modified. If the user surface is replaced by another human/computer interface, such as voice activated command input, only the code that maps the user input into the intermediate language needs to be modified.

The communication system (Bienz and Hunter, 1988) permits teleoperation by $9600baud$ dialup modem, a Sytek data communication network, or an Ethernet data communication network.

## Observations

(1) Analytical work in the life sciences or microgravity sciences cannot proceed in an economically feasible fashion without facilities for automated remote fluid handling. The work is tedious and repetitive, and crew time is exceedingly expensive. Thus it is essential that the telescience principles developed in this testbed be extended and integrated with other activities such as the Space Station Life Sciences Glovebox. In particular, an instrument rack (automated laboratory) should be immediately added to the design, adjacent to the glovebox, with provision for passing samples back and forth between the two areas.

(2) The robot configuration must be geometrically compatible with the workspace. For example, in front of a flat rack, it is inappropriate to use a robot that is unable to move horizontally and vertically along the rack. It is insufficient that we are able to reach each point in space somehow. Unless the robot configuration matches the workspace topology, the necessitated indirect approximation of the desired robot motion is paid

for by an unavoidable loss in positioning accuracy and by unnecessary complexity of the command sequences.

(3) The handling of small amounts of fluid in space calls for special equipment (syringes or pipettes). The laboratory robot must be able to handle these devices. Today's commercial laboratory robots are not equipped with an appropriate end effector for that purpose. As the variety and complexity of the tasks to be performed by the robot increases, special attention must be paid to quick–change end effectors and dexterous manipulation.

(4) The major fluid handling problems which must be solved relate to contamination control and waste disposal, coupled with the fact that there can be no air/liquid interfaces which are not completely controlled with surface tension.

**Achievements**

(1) Our testbed has produced the design of an appropriate end effector / syringe coupling. Several alternatives were investigated, and several alternatives were actually built. This approach works, but is somewhat cumbersome in terms of detailed steps required in the robot programming, and in the amount of waste produced (a new syringe is needed for each operation).

(2) We are currently experimenting with a setup using a motorized pipette. In this approach, only the cheap, light, and small plastic pipette tip needs to be discarded. All other parts are reusable. This configuration seems also in line with industrial developments on the ground. In 1984 alone, industry consumed 5 billions of these pipette tips. It will be necessary to modify the tip to provide containment/insertion for contamination control, and to avoid air/liquid interfaces. Our testbed will help to decide whether or not this technology will work satisfactorily for telescience operations in microgravity environments.

(3) Successful teleoperation of a representative (albeit small scale) fluid handling laboratory has been demonstrated. We have been successful in producing generic human/computer and computer/computer interfaces which work equally well for teleoperation of two quite different demonstrations. This could not have been accomplished as quickly and elegantly without the prior development of the OASIS software by the University of Colorado.

(4) It was established that color video feedback is a *conditio sine qua non* for teleoperation of a robot controlled fluid handling laboratory. We have determined that the recommended video data compression technique for the robot/laboratory observation data is one that currently has wide–spread use in the area of videoconferencing. Since color images are needed for this application, data rates of 80 to $400 kbits/sec$ (depending on rates of motion) are required for adequate image quality (Tolle, 1988).

(5) It was determined that man–in–the–loop remote control of the robot will not work satisfactorily if the round trip delay time (consisting of communication delays and processing delays for both the uplink and the downlink) is longer than approximately $1 sec$. Under normal operation, such a short delay time cannot be maintained. The current thinking in the Space Station community is such that a round trip delay time of $5 sec$ can be reasonably expected, but end–to–end integration real–time performance simulation of the Space Station Information System (SSIS) including the communication links (through the TDRSS satellite) must be used to verify this unproven hypothesis. It was determined that, under normal circumstances, a delay time of $5 sec$ will suffice if the robot is operated with local automatic control circuitry, whereby the remote commander is limited to determining the set points. That is, the remote commander can tell the robot to extract $0.5 ml$ of liquid from container $B$, but cannot step the robot manually to the desired position. It is suggested that NASA provide emergency channels that can be accessed for a limited time period after something went wrong,

for example to get a jammed robot back on track. These emergency channels should guarantee a round trip delay time of less than 1*sec* (Pan, 1987).

(6) The varying and unpredictable time delays associated with public packet switched communication channels (for example using the NSF backbone) will jeopardize any reliable teleoperation of robots aboard the SSF from the ground since it is questionable that even the 5*sec* time delay that we talked about above can be guaranteed under this mode of operation. To verify this statement, it is vitally important that a realistic end–to–end integration real–time performance simulation of the SSIS be performed quickly.

## DEVELOPMENT OF GENERIC TECHNOLOGY FOR TELEOPERATION

As previously noted, we implemented two seemingly quite different testbed demonstrations. These were selected in part so that a generic set of technologies for teleoperation could be investigated. This component (i.e. the commonality between teleoperation of seemingly very different experiments aboard Space Station) is being reported separately to emphasize the various aspects of generic Telescience technology development.

### Method of Investigation

The two scientific experiments, teleoperation of an astrometric telescope and teleoperation of a fluid handling laboratory, are quite different from a point of view of the equipment to be controlled and the scientific data and telemetry required, but a very similar hardware/software architecture was used for both (see Fig.3).

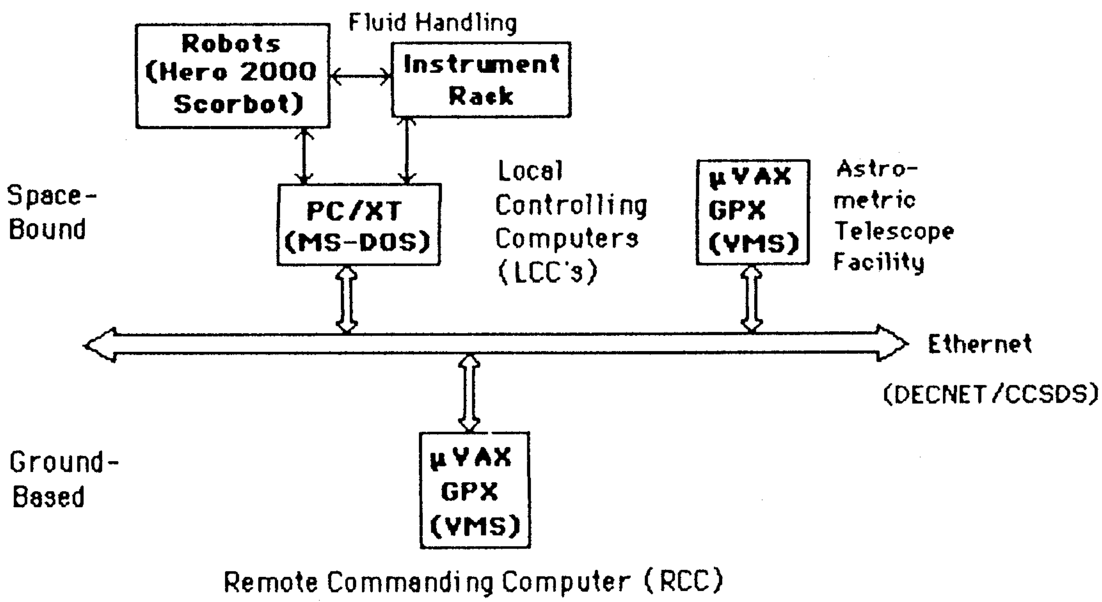


**Figure 3.** University of Arizona Telescience Laboratory

In both cases, the architecture consists of a remote commanding computer (RCC) which communicates at 9600*baud* over dialup phone lines, a Sytek network, or an Ethernet network with a local controlling computer (LCC). In both cases, the RCC consists of a $\mu$Vax–II/GPX workstation which houses a human/computer interface consisting of an application of the Operations and Science Instrument System (OASIS) (Hansen, 1987). For the astrometric telescope, the LCC is a second $\mu$Vax workstation which also runs the telescope simulation. For the fluid handling laboratory, the LCC is an IBM PC compatible with 640$K$ memory, an 8087 coprocessor, a 20*Mbyte* hard disc, and a LabTender multifunction board. For both experiments, intermediate command language statements, telemetry, and scientific data are exchanged using DECnet and CCSDS packets as the communication protocols (Bienz and Hunter, 1988).

Ultimately, we would like to use free–syntax plain English for expressing user directives. We are currently far from this ultimate goal. However, OASIS presents us with an excellent compromise. While the number of directives that can be understood by OASIS is fairly limited in every application, OASIS supports the creation of user interfaces for new applications in a rather convenient manner. The user interface is entirely data driven (through an application data base), and it takes an experienced programmer a fairly short time to create a data base for a new application including the design of appropriate windows and icons (depending on the complexity of the application between one and four weeks). The user interface could further be improved by offering an interactive application data base creation module. This could reduce the time needed to generate a new application data base to a few hours.

Thus, while OASIS has been designed to interact with experiments directly, its true strength lies in its user interface capabilities. The current software is a little slow in performing these actions due to the fact that the application interface is completely data driven. Speed and flexibility are always in strong competition with each other. We suggest that it would be a good idea for the University of Colorado to look into the possibilities of creating a data compiler that can be used to compile a new application data base into a set of Ada procedures that can be linked with the OASIS software once the new application data base has been completely debugged and tested. This could speed up the execution of OASIS by a factor of 10 to 100. In this way, the best of both worlds (flexibility of the data driven approach, and execution speed of the code driven approach) can be combined in an optimal manner at the expense of one (very slow but insignificant) process of compilation and linkage once the new application data base has been implemented and thoroughly tested.

There will arise the need for an AI planner. This need can be explained as follows: The user might wish to tell the robot to *"go and paint my car green"*. However, the robot is probably not able to understand such a high level directive. For this purpose, we would need an algorithm that maps high level directives into (eventually an extensive) series of lower level commands that can be understood by the robot. Currently, OASIS does not provide for this capability at all. The (semantic) actions to be taken as a result of a (syntactic) user directive are specified in OASIS in the application data base in the form of an attributive grammar with semantic rules attached to the syntactic commands. This method does not provide for the necessary flexibility since in many applications, there does not exist a unique mapping between the high level directive and the lower level commands. In our example, the robot could use a brush or a spray, he could start from the rear or from the front, and finally, he could sit in the car, drive it to the next garage, and subcontract the task.

OASIS can perform a limited set of tasks related to the LCC, but the software is not particularly strong in this respect. For instance, it seems obvious that we may wish to initiate several actions simultaneously as long as they do not conflict with each other. OASIS allows to process only one command at any given time.

Moreover, it is not meaningful to combine all the activities related to the remote control of an experiment in one program since, in reality, the Space Station software will be distributed between the end user's Workstation (on the ground), and his experiment (aboard Space Station) with an uplink somewhere in between. It would be useful to modularize OASIS for distributed processing. OASIS allows us to do this to a limited extent, but is fairly weak in its communication capabilities. For example, it would be useful if OASIS could be told to expect an acknowledgment of a transmitted command from the receiver, and in case of timeout, resubmit the command. OASIS currently won't do this.

In our testbeds, we ignored OASIS' capabilities to act as an LCC, and split the task between two separate computers, the Remote Commanding Computer (RCC) which runs OASIS, and the Local Controlling Computer (LCC) which runs our own Ada–coded LCC programs. This approach allowed us to use OASIS for what it can really do well, namely communicate with the user, and it allowed us to study problems that relate to the distributed nature of the overall SSIS hardware/software architecture. However, in the future, OASIS could be enhanced in such a way that it could be used on both ends of the communication chain, i.e., as a user interface program AND as an experiment interface program. Both programs would be completely data driven (through application data bases), and the communication between the two involved OASIS programs could be made completely transparent to the user. It would also be possible to include mechanisms which would allow multiple RCC's to communicate with one or multiple LCC's using the same communication network. Indeed, there is no fundamental difference (from a computer program point of view) between the human operator and the experiment to be controlled.

### Experimental Results

The goal of developing generic modular software for teleoperations was attained. Our conclusions, recommendations, and suggestions for future work are as follows:

(1) **Scientists will require remote multiple access for teleoperation.** While a number of TTPP participants have looked into the problem of remote multiple access to distributed data bases, little has been done with respect to multiple control of experiments. Outside of the telescience community, there exists even less awareness of the need for multiple simultaneous users of the same equipment. We have performed a feasibility study which indicates that multiple simultaneous control and/or observation of an ongoing experiment can be attained.

(2) **The need for multiple access must be addressed now.** It is necessary to foresee multiple simultaneous controllers/observers right from the beginning since this mode of operation calls for additional communication protocols and equipment. For example, if we want to allow an observer to obtain on his screen a shadow image of everything that is displayed on the main experimenter's console, all screen commands, including the house–keeping commands such as changing the background color of a screen window, must be communicable rather than being treated as strictly local activities. A more obvious example is the use of token–passing (key ownership) to ensure that only one location is in active control at any particular time. It is therefore not feasible that the system be developed for single users only, with the design of a multiple user capability, for financial and scheduling reasons, being postponed until later. It is important that this capability be foreseen right away.

(3) **Multiple experiments must also be considered.** Due to a lack of appropriate coordinating information, all activities are currently treated as independent units. The general scenario is that of an experiment on one end remotely controlled by an experimenter on the other end. In reality though, all communication to and from Space Station will be routed through a central (though distributed) operating environment (the Operation Management System (OMS)). The integration of the telescience activities into the OMS will create additional problems (such as additional time delay), and will call for additional communication protocols. It is important that the implications of this integration into the OMS be considered early on. It might be useful to create an OMS simulator, and make this simulator available to the telescience community. The TTPP participants will then be able to route their commands through the OMS simulator to more accurately determine what will happen to their experiments aboard the SSF.

(4) **Overall system response times must be specified.** We have studied the effect of communication delays on stability of remote closed loop control. It appears that a total delay of $1sec$ can be made acceptable by use of special techniques, but much more than that will result in unstable behavior of the control circuit. For reference, the TDRSS hop alone introduces a communication round trip delay of approximately $0.4sec$. Since we will not be able to guarantee $1sec$ delay for normal operation, it will not be possible to routinely teleoperate any robots with a remote man–in–the–loop control configuration. Instead, the local robot control must be automated. The remote operator cannot control the robot directly. His normal mode of interaction must be limited to specifying set points and triggering the execution of canned experiments (using preprogrammed procedures). If something goes wrong, however, it must be possible to remotely fix the problem. For that purpose, it should be possible to temporarily request a point to point connection (special communication link) which guarantees a delay, including processing time, of less than $1sec$ round trip ($0.5sec$ per path) For normal operation (specification of set points or procedures), our testbed has shown that a response time of $5sec$ will be sufficient, or rather, that the control circuits can be designed to operate under such conditions.

(5) **Delay times must be verified by simulation.** It is important to study the effect of these multiple software and communication layers on overall system response time. Since the development of these software tools has been distributed among several contractors (the Software Support Environment (SSE) and the Work Package II (WP2) contractors), nobody seems to have a clear picture of the effect of these multiple software systems on overall system response times. It is suggested that an integrated end–to–end real–time performance simulator be created to study this problem.

(6) **Public data switching networks cannot be used for teleoperation.** The performance available through unlimited access data switching networks (such as NSN/NSF) won't suffice under any conditions. The large, unpredictable, and variable delay times imposed by these networks will jeopardize any successful teleoperation of equipment aboard Space Station (the problem is with the delay, not with the bandwidth — NSN will probably work fine for remote access to distributed data bases). There are those who claim that this problem will be eliminated by installation of higher capacity networks, but this is not true unless access is limited. As an example of this, the NSFnet backbone was upgraded from $56kbps$ to $448kbps$ last July, and it is planned to increase its capacity to $1.54Mbps$ in 1989 and $45Mbps$ in the early 1990's. The problem is that usage is also increasing (100% per year) so that the current grade of service is not predicted to improve much. This situation could be altered with provision of a virtual circuit capability, but this seems far in the future, if it occurs at all.

(7) **Video feedback has separate requirements.** Bandwidth AND delay problems occur with the incorporation of video feedback. It is suggested that this problem be considered outside the OMS. Special point to point connections with a guaranteed delay time of below $2.5sec$ must be established which bypass the OMS and can deliver video feedback directly to the teleoperator. More research is needed to establish appropriate data links, switching control, and data reduction schemes for different applications.

(8) **Ada is an excellent language choice for telescience.** Ada has proven to be an excellent choice as the standard programming language! In particular, it is a very convenient language for telescience applications. Especially useful are the *exception handling* capabilities which allow separation of the error handling from the processing of correct commands (increased readability and modularity), the *information hiding* features (private types) which allow for a previously never seen ability to modularize software, and to distribute the coding of subsystems among several team members without fear of side–effects, and finally the *multi–tasking capability* which allows capture of parallel processes in parallel software modules in a very convenient manner.

(9) **OASIS should be modified and extended.** We have extensively tested OASIS for the purpose of teleoperation of experiments aboard Space Station. We found that OASIS presents us with a flexible and convenient state–of–the–art user interface to command experiments. However, OASIS is not yet satisfactory with respect to its flexibility and performance for communicating with the equipment located at the other end of the communication link, and for interactions of multiple experimenters with multiple experiments.

## SUMMARY AND CONCLUSIONS

This paper has described the results of two scientific experiments and an underlying technology experiment performed by the University of Arizona, College of Engineering and Mines, as part of the Telescience Testbed Pilot Program (Schooley and Cellier, 1988a, 1988b; Schooley *et al* 1989). The first experiment involved teleoperation of a forerunner of the Astrometric Telescope Facility, a proposed attached payload of the Space Station Freedom, and the second involved teleoperation of a remote fluid handling laboratory for use with life science or microgravity science experiments.

The telescience approach to scientific investigations in remote or dangerous locations has been validated. The general objectives of less crew time, more science, and better science can be attained through this approach. Scientific productivity was markedly increased, not only in our testbeds, but those of the other universities as well.

This achievement has been made possible by recent technology advances in communications systems, control systems, computers, remote vision and sensing, visual displays, and robotics. These technologies are sufficiently mature that telescience concepts can be included in all future missions, but additional research is required to ensure operational reliability, and to fully exploit the advantages of these new techniques.

## ACKNOWLEDGMENTS

We wish to thank Elaine Hansen and her staff of the Laboratory for Atmospheric and Space Physics (LASP) of the University of Colorado for a cordial cooperation, and for their effective and efficient support of our research efforts. Without given access to their Operations and Science Instrument System (OASIS), we could not have obtained the results that are presented in this paper.

## REFERENCES

Bienz, R.A., and J.J. Hunter (1988), *Communication Software Design for Telescience Demonstrations*, Technical Report TSL–019/88, Electrical and Computer Engineering Department, University of Arizona, Tucson AZ.

Cellier, F.E. (1987), *Teleoperation of the Thaw Telescope at the Allegheny Observatory: A Case Study*, Technical Report TSL–004/87, Electrical and Computer Engineering Department, University of Arizona, Tucson AZ.

Hack, B.W.J. (1988), *Man to Machine, Machine to Machine and Machine to Instrument Interfaces for Teleoperation of a Fluid Handling Laboratory*, MS Thesis, Technical Report TSL–014/88, Electrical and Computer Engineering Department, University of Arizona, Tucson AZ.

Hansen, E. (1987), *OASIS – Operations and Science Instrument Support, System Manager's Guide, Database Guide, CSTOL Reference Manual*, Laboratory for Atmospheric and Space Physics (LASP), University of Colorado, Bolder CO.

Lew, A.K. (1988), *Astrometric Telescope Simulator for the Design and Development of Telescope Teleoperation*, MS Thesis, Technical Report TSL–016/88, Electrical and Computer Engineering Department, University of Arizona, Tucson AZ.

Pan, Y. (1987), *Teleoperation of Mechanical Manipulators Aboard the U.S. Space Station*, MS Thesis, Technical Report TSL–002/87, Electrical and Computer Engineering Department, University of Arizona, Tucson AZ.

Pan, Y., and A.K. Lew (1988), *Teleoperations Software for Remote Fluid Handling*, Technical Report TSL–020/88, Electrical and Computer Engineering Department, University of Arizona, Tucson AZ.

Raize, E. (1988), *Syringe Driver Assembly for Automated Fluid Handling Laboratory*, Technical Report, TSL–012/88, Electrical and Computer Engineering Department, University of Arizona, Tucson AZ.

Schooley, L.C., and F.E. Cellier (1988a), *Telescience Testbed Pilot Program Final Report*, Technical Report TSL–021/88, Electrical and Computer Engineering Department, University of Arizona, Tucson AZ.

Schooley, L.C., and F.E. Cellier (1988b), *Teleoperation of a Simulated Astrometric Telescope*, Technical Report TSL–022/88 (videotape), Electrical and Computer Engineering Department, University of Arizona, Tucson AZ.

Schooley, L.C., F.E. Cellier, and D.G. Schultz (1989), *Teleoperation of a Fluid Handling Laboratory*, Technical Report TSL–023/89 (videotape), Electrical and Computer Engineering Department, University of Arizona, Tucson AZ.

Walker, W.T. (1988), *Video Data Compression for Telescience*, MS Thesis, Technical Report TSL–017/88, Electrical and Computer Engineering Department, University of Arizona, Tucson AZ.