The Space Congress® Proceedings

2003 (40th) Linking the Past to the Future - A Celebration of Space

May 1st, 1:30 PM - 4:30 PM

Paper Session II-C - Data Access and Procedure Generation for the International Space Station

Dawn M. McIntosh
NASA

Sharif Elcott
QSS Group, Inc.

Bradley J. Betts
Computer Sciences Corporation

Robert W. Mah
NASA

Procedure Visualization to Augment Space Mission Training

Dawn M. McIntosh[1], Sharif Elcott[2], Bradley J. Betts[3], Robert W. Mah[1]

Smart Systems Research Laboratory
NASA Ames Research Center, M/S 269-1
Moffett Field, CA 94035-1000

[1]NASA, {Dawn.M.McIntosh, Robert.W.Mah}@nasa.gov
[2]QSS Group, Inc., sharif@email.arc.nasa.gov
[3]Computer Sciences Corporation, bbetts@email.arc.nasa.gov

Abstract

The Intelligent Virtual Station (IVS) has been developed by the Smart Systems Research Laboratory at the NASA Ames Research Center as a solution to some of the training and operations challenges faced by organizations like the International Space Station training facilities and Mission Control engineering teams. At present, astronaut crews are constrained by limited access to physical mockups, which themselves have a built-in 1-g limitation. Mission operations teams are faced with the daunting task of controlling the operations and maintenance of an ever-changing Station in space. Many operations teams create and follow textual procedures without the ability to visualize the given actions or alternatives. The IVS allows users to easily generate and view procedures to enhance training and operations. Because training and mission operations are of crucial importance to the International Space Station and other similarly sophisticated programs, this paper is focused on the IVS integrated procedure tool.

1. Introduction

The International Space Station (ISS) is a very complex orbiting research facility that is being designed and built by sixteen different nations. Adding to the challenge are the problems associated with assembling the Station in the hostile environment of space. Given its complexity, it is not surprising that NASA and its international partners are looking for ways to improve the training of astronaut crews that will live aboard the ISS and the mission ground controllers that will monitor and control its operation. New technologies are sought that can bring verifiable benefits to the task of training, such as minimizing the time required to train for a task, increasing knowledge retention, and gaining new insight into the operation of the station.

The Intelligent Virtual Station (IVS) is a new approach being developed to afford complex systems such as the ISS an ever-growing solution to some of the difficult challenges facing training and operations. This includes making pertinent information easily accessible during training and simulation [1], providing spatial awareness through the use of non-immersive virtual environments (VEs) [2], developing software frameworks to support the addition and integration of specialized tools to meet a variety of training and operations needs [3], and generating and visualizing critical spacecraft procedures such as the replacement of on-orbit equipment (known as Orbital Replacement Units). This paper is focused on this last area—authoring and visualizing spacecraft procedures—for the ISS. The work done to date, while believed to be more broadly applicable, has been focused on the ISS.

The IVS offers an intuitive single interface for the crew to interact and participate in more realistic and advanced training scenarios. Both crew and crew training teams can generate procedures involving the manipulation of objects using a non-immersive virtual environment running on a portable computer. This capability applies to both scientific-based procedures and maintenance/repair procedures. Astronaut crews currently train for these types of procedures by reading textual descriptions of the task, and then practicing the procedure in a physical hardware mockup facility with their trainers. With the IVS, crews would be able to view a procedure created by a trainer or scientist, then practice the procedures using the IVS before moving to the physical mockup. Moreover, crewmembers can repeat the procedure training on their own laptops at their convenience as many times as desired prior to their mission. The procedure generation and viewing tool can also be useful for mission operations engineers and ground controllers.

They can easily create procedures in real-time to handle unplanned events or emergencies.

2. IVS Overview

Before presenting the details of procedure generation and visualization, a brief overview of some of the features of the IVS will be given; a more detailed presentation can be found in [2]. There are three main feature sets in the IVS: a 3D virtual environment, a data management system, and simulation and decision making tools. The virtual environment provides a detailed model of the ISS, with model geometry repurposed from CAD into a format suitable for real-time rendering on portable computers. The information management system accesses local and remote databases containing ISS document-type data. Finally, the simulation and decision making tools are at present divided into two areas: one interfaces with existing NASA training systems, and the other provides procedure generation and visualization. Future work will look to expand the number of areas to include fault detection and identification, health monitoring, and tools designed to improve decision making.

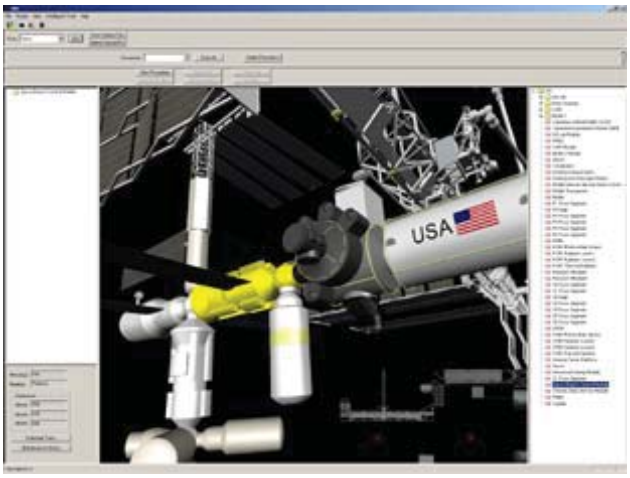

Figure 1 Screen shot of the IVS application. The Zarya module has been selected by the user.

Figure 1 is a screen shot of the IVS running on a laptop computer (Dell Latitude C840, Windows XP, Pentium 4 at 1.7 GHz, 512 MB of RAM, NVIDIA GeForce4 440 Go graphics card). The center pane is the non-immersive VE displaying a 3D graphics model of the ISS. In this case, the user has navigated to an exterior viewing position of the ISS (the ISS is shown in its nominal final assembly configuration). The user has selected the Russian Zarya module for use in a procedure. The bottom right corner of the center pane shows a navigation guide map of the station. The map provides top and frontal projections of the ISS with the users current viewing position superimposed, thereby allowing the user to determine his/her current position with respect to the station. The left and right panes are referred to respectively as the document explorer and the component explorer. Both have the look and feel of conventional file explorers. Objects in the VE can be selected, either by clicking on the object directly in the VE or by using the component explorer. That action in turn brings up data associated with the object in the document explorer. The upper panes contain the user interface to the procedure visualization tool, the subject of the next section of this paper.

The IVS is developed in C++ under Microsoft Visual Studio .NET; it has approximately eighty core classes and thirty-five thousand lines of code. It deploys on Windows computers in either a networked or non-networked setting (in the latter setting, resources such as remote databases are obviously not accessible). The software toolkits used include Microsoft Foundation Classes [4] and OpenGL [5, 6].

3. Procedure Tool

The procedure visualization tool consists of two distinct components: one that allows procedures to be generated (an authoring capability), and the other that allows procedures once created to be viewed. An important design requirement is to make the tool easy to use to maximize training effectiveness and efficiency.

For the procedure generation component, the user simply selects the object of interest, initiates the creation of a new procedure, and chooses to translate (and possibly rotate) the object. Multiple objects can be serially selected and manipulated in this fashion during a single procedure generation session. During translation, the user can set positions defining a point in Euclidian world space that the object is required to move through. This is important when the object must avoid another object or move around a corner. Setting positions that the object must traverse through allows the user to create desired pathways or build in object avoidance. Users may choose either a curve-smoothing polynomial interpolation or a piecewise linear spline for each trajectory step in the procedure. When the object is to translate through multiple locations in the interior space of a module, a smoothed trajectory path has a more realistic look. In some cases, such as pulling an object out of the rack, a straight line trajectory is of course required. Rotation trajectories are also smoothed to provide a more realistic simulation for the procedure.

At the class level, a `Procedure` is composed of a vector of `Path`s. Each `Path` is a line or curve through Euclidian world space that a particular `Object` must follow, where an `Object` is a geometrical object

displayed in the VE. A `Procedure` is set in motion via a `Go` message. This message is propagated to each `Path` in turn. At a given instant in time, the `Path` determines the location and rotation appropriate for its `Object` and then instructs the `Object` to render itself. Figure 2 provides a UML object diagram of the various classes involved; to reduce clutter, only portions of the public interface and attributes relevant to procedures are shown.

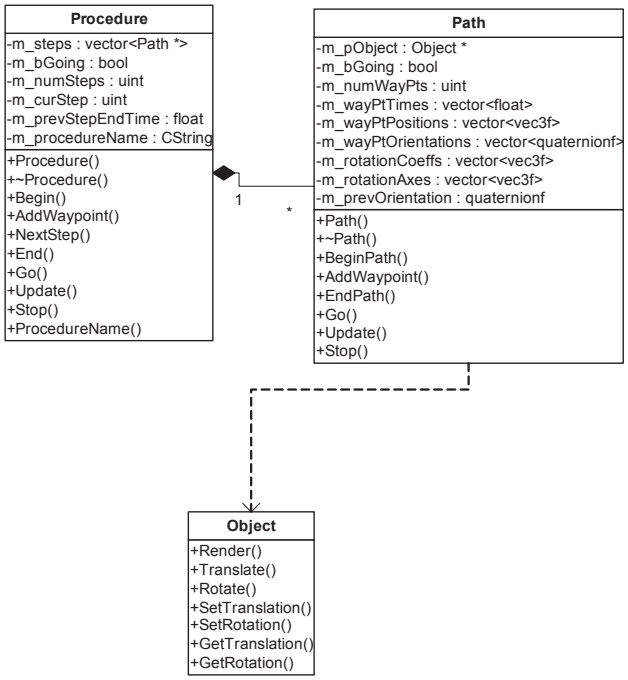

Figure 2 UML object diagram of core classes involved in procedure generation and viewing.

Once a procedure has been generated, the Procedure object is saved locally to disk (using standard serialization methods inherited from the MFC `CObject` class). The procedure file can be shared with other IVS users who could then view the newly created procedure. Figure 3 shows a user authoring a procedure in which a particular object (the Life Science Glovebox) is being translated.

The viewing component of the IVS procedure tool allows a user to select and run a locally stored procedure. Unlike a standard movie file, IVS users have the ability to navigate to any viewpoint during the procedure playback. A limitation of the procedure playback is that the translation and rotation rates are fixed and cannot be modified by the user. It may be important for a user to show an object spinning quickly, or translating slowly when passing in close proximity to delicate instruments that could be damaged if contacted. Future versions of IVS will offer the user the ability to select and modify these rates.

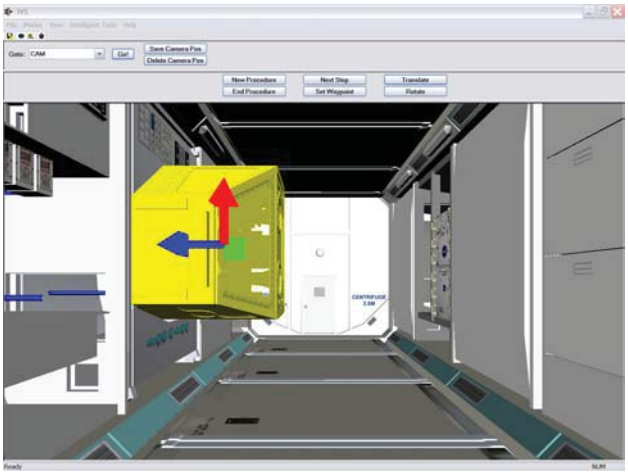

Figure 3 Screen shot of a user authoring a procedure.

4. Related Work

The need to visualize and assess training and operations procedures is critically important for many complex systems. Several tools for procedure visualization are available. The IVS allows users to easily generate and visualize procedures for both planned and unplanned tasks. Some of the existing tools provide the users with an interactive viewing GUI [7], and allow the user to select individual steps as well as the ability to view relevant text [8]. DELMIA's high-end product, ENVISION/ERGO [9], has an interactive interface for viewing procedures, provides users with an interface for generating procedures, and offers motion capture, collision detection, and cycle time derivation. A DELMIA plug-in product is required to play back procedure simulations. Existing tools simulate procedures performed in the 1-g environment only.

Many aerospace companies and automobile manufacturers use tools such as those listed above in their design, development, and assembly phases. In recent years, many of these companies have turned to virtual reality to achieve higher levels of realism and analysis efficiency.

Virtual reality (VR) has been widely used in the world of construction. It is being used to train construction workers on the operation of equipment and construction techniques. One VR tool, VECWIT [10], combines a 3D virtual construction site with other visual media to convey information on safety, load weight, and distances. Assembly and disassembly procedures are practiced in the VR environment with feedback to the user if they failed to accomplish the task.

The 3D simulation tool COSIMIR® [11] and its variants are also used to satisfy many types of training needs. COSIMIR® VR is a variant of the COSIMIR® tool being used by the German Space Agency for training astronauts and as a planning tool for conducting robot-assisted experiments [12, 13]. The COSIMER® VR tool is used to train astronauts for the ISS COLUMBUS module in a distributed environment [12]. In this application, the user can practice a set of actions prior to performing the actual experiment with robotic tools [13].

5. Conclusions and Future Work

This work has illustrated a straightforward technique for generating procedures for complex systems. One conclusion, and a novel feature of the IVS, is that the ability to author procedures within a VE easily and in real-time can be done with a small set of interacting classes. As well, using standard class serialization techniques dramatically reduced the effort involved in storing and sharing procedures (at the expense of human readability). Another conclusion drawn from this work is the necessity of working early and often with potential users. Their involvement has and will continue to be useful in identifying critical research challenges. Finally, as the survey of related work shows, VR training, while a still-maturing field, has a broad range of disciplines to which it can be applied.

There is a need to augment current training due to limitations posed by text-based procedures and physical mockups. With that in mind, the future development of the procedure tool will focus on the following:

- An enriched feature set, including the ability to: reuse procedure steps; allow connections between objects; apply deformations to hoses and wires; allow conditional branching; dynamically modify training scenarios; include textual information in a procedure step to accommodate non-physical actions; and vary rotation and translation speeds.
- Offering collaborative VE services, including sharing control of the generation of a procedure.
- Procedure generation with dynamic modeling in the 0-g environment and high-resolution collision detection.
- Procedure learning features to achieve more realism, including using time data from procedure segments actually performed in space.
- Procedure scheduling features to factor in crew time constraints or to determine the feasibility of performing complex tasks.
- Automatic conversion of an authored procedure to a text document.

Potential applications of the IVS procedure tool will include using it to evaluate and optimize task procedures before they are approved for use in astronaut crew training, using the procedure tool to help visualize and assess clearance between moving objects, and evaluating human factors issues associated with crew maneuvers through the ISS modules.

6. Acknowledgements

The authors gratefully acknowledge the efforts and contributions of the other IVS developers: Richard Papasin, Rommel del Mundo, Dr. Edward Wilson, Mike Guerrero, and Brian Niehaus.

This work was funded by the NASA Computing, Information and Communications Technology Program under the Computing, Networking and Information Systems Project.

7. References

[1] B.J. Betts et al., "A Data Management System for International Space Station Simulation Tools," *Proc. 2002 Int'l Conf. App. Modeling and Simulation* (AMS 2002), ACTA Press, Boston, MA, Nov. 2002, pp. 500-504.

[2] R. Papasin et al., "Intelligent Virtual Station," to appear in *Proc. 7th Int'l Sym. Art. Intell.* (iSAIRAS 2003), Nara, Japan, May 2003.

[3] B.J. Betts et al., "A Software Framework to Enhance Training and Operations of Space Missions," to appear in *Proc. Space Mission Challenges for Info. Tech.* (SMC-IT 2003), Pasadena, CA, Jul. 2003.

[4] M. Williams et al., *Visual C++ 6 Unleashed,* Sams Publishing, Indianapolis, IN, 2000.

[5] OpenGL Home Page, http://www.opengl.org (verified Apr. 2003).

[6] M.Woo et al., *OpenGL Programming Guide: The Official Guide to Learning OpenGL, Version 1.2,* 3rd ed., Addison-Wesley, Reading, MA, 1999.

[7] http://www.xvl3d.com/en/demo/indexh.htm (verified Apr. 2003)

[8] http://www.actify.com/v2/products/SFViewer/webshowcase.htm (verified Apr. 2003)

[9] http://www.delmia.com/gallery/pdf/ergo.pdf (verified Apr. 2003)

[10] J. Assfalg, A. Del Bimbo, and E. Vicario, "Using 3D and ancillary media to train construction workers," *IEEE Multimedia*, vol. 9, no. 2, 2002, pp. 88-92.

[11] E. Freund and D.H. Pensky, "COSIMIR® Factory: Extending the Use of Manufacturing Simulations," *Proc. 2002 Int'l Conf. Robotics & Automation,* Washington, D.C., May 2002, pp. 2805-2810.

[12] E. Freund and J. Rossman, "Rapid Prototyping, Astronaut Training and Experiment Control and Supervision: Distributed Virtual Worlds for COLUMBUS, the European Space Laboratory Module," *Proc. SPIE: Telemanipulator Telepresence Tech. VIII,* Newton, MA, vol. 4570, 2001, pp. 113-122.

[13] E. Freund, J. Rossmann, M. Schluse, "Projective Virtual Reality in Space Applications: A Telerobotic Ground Station for a Space Mission," *Proc. SPIE: Sensor Fusion and Decentralized Control in Robotic Systems III,* vol. 4196, 2000, pp. 279-290.