Proceeding of the 7th International Symposium on Artificial Intelligence, Robotics and Automation in Space: i-SAIRAS 2003, NARA, Japan, May 19-23, 2003

Intelligent Virtual Station

Richard Papasin[1], Bradley J. Betts[2], Rommel Del Mundo[3], Michael Guerrero[4], Robert W. Mah[1], Dawn M. McIntosh[1], Edward Wilson[5]

Smart Systems Research Laboratory
NASA Ames Research Center
M/S 269-2, Moffett Field, CA 94035-1000, U.S.A.

[1]NASA, {Richard.I.Papasin, Robert.W.Mah, Dawn.M.McIntosh}@nasa.gov
[2]Computer Sciences Corporation, bbetts@email.arc.nasa.gov
[3]QSS Group, Inc., rdelmundo@mail.arc.nasa.gov
[4]Guerrero Engineering, mguerrer@email.arc.nasa.gov
[5]Intellization, ed.wilson@intellization.com

Keywords information management, virtual environments, virtual training, virtual operations

Abstract

The Smart Systems Research Laboratory (SSRL) at the NASA Ames Research Center has developed an Intelligent Virtual Station (IVS) prototype for the International Space Station. The IVS addresses major challenges faced by astronauts, mission controllers, and engineers such as efficiently managing large amounts of information. It also targets certain limitations in virtual training and operations. IVS offers a potential solution and enhances capabilities to bring disparate sources of data together in a single unified interface. The IVS also offers a software platform to build and extend various interfaces for users in the context of virtual training and operations. An overview of the IVS, its goals, and the approach taken is presented in this paper. Brief implementation details of the IVS are also provided.

1. Introduction and Motivation

The International Space Station is a collaborative endeavor among sixteen nations to build the largest and most complex space-based scientific research facility yet seen [1]. Its complexity poses major challenges to astronauts, mission controllers, and the various engineering teams working on it with regard to training and operations. One of the major challenges is dealing with large amounts of information associated with the ISS. With many international partners working cooperatively on designing, building, and assembling the ISS, an overwhelming amount of data is generated and accessed. These include preliminary and critical design review documents, schematics and diagrams, operations manuals, engineering analysis data, problem reports, and procedures, to list but a few. The documents are typically stored remotely in various databases, websites, or file servers.

The Intelligent Virtual Station (IVS) offers the capability to bring these disparate sources of data together in an intuitive single unified interface. Physical components in the IVS are represented in a three-dimensional virtual environment and by two-dimensional file hierarchies. A user can simply click on components in the virtual environment to view, access, and retrieve associated information. This allows users to focus more on understanding, utilizing, and analyzing their data than on retrieving it. A screenshot of the IVS application is shown in

This paper is in part authored by employees of the U.S. Government and is in the public domain. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the U.S. Government. Work was funded by the NASA Computing, Information and Communications Technology Program (CICT) under the Computing, Networking and Information Systems Project (CNIS).

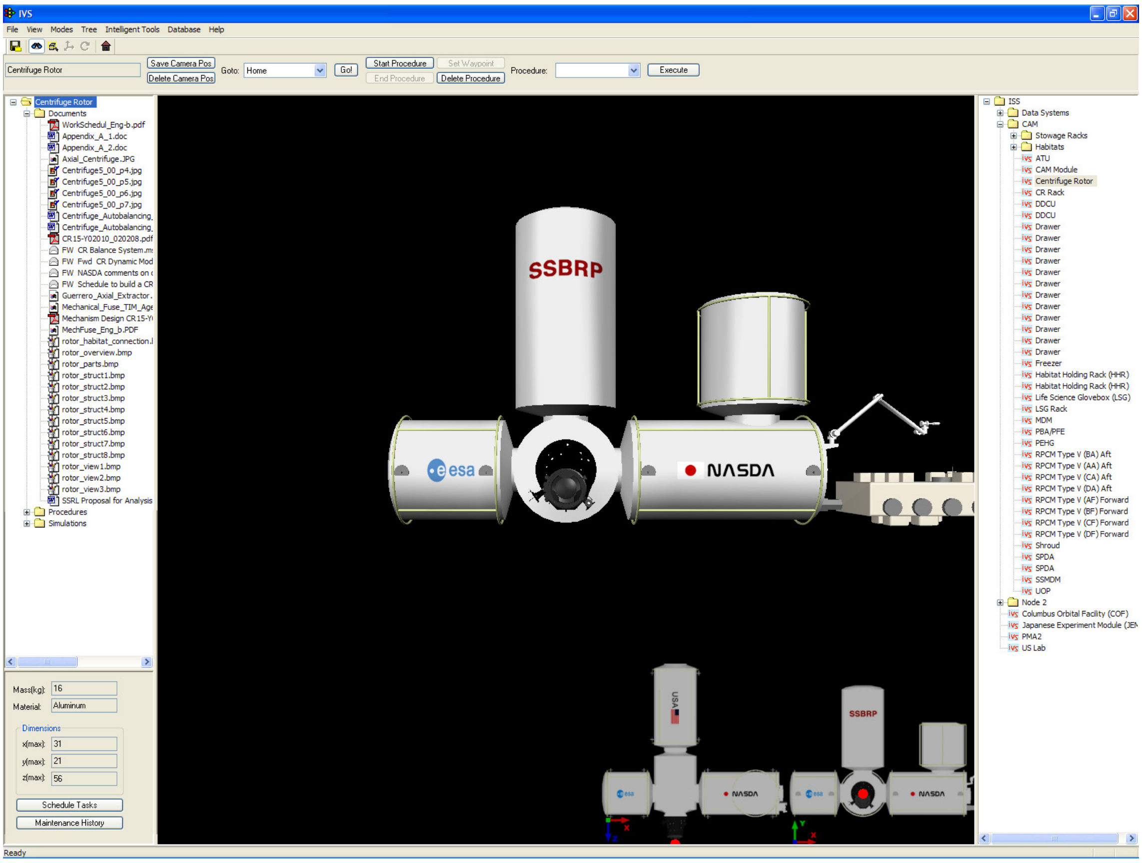

Figure 1: Screenshot of the Intelligent Virtual Station.

Figure 1. In the center pane of the figure is the 3D rendering component. Also in the center pane, in the bottom right corner, is a guidance map to determine location and facilitate navigation in the station. The left and right panes are respectively referred to as the document explorer and the component explorer.

Another major challenge faced by ISS teams is training for and operating the complex orbiting facility. Astronauts and mission controllers train for and operate the International Space Station predominantly from facilities at the NASA Johnson Space Center. Facilities such as the MCC (Mission Control Center) and the SSTF (Space Station Training Facility) play significant roles in the operations and training of the crew for the ISS. While extremely effective, the physical natures of these facilities pose some limitations. For example, astronauts should be able to participate in meaningful SSTF training while at remote locations. Crew members are often stationed for long periods of time at places like the NASA Kennedy Space Center in Florida, the European Astronaut Center in Cologne, Germany, and the Cosmonaut Training Center in Star City, Russia. While there, remote training options are very limited or nonexistent. The IVS provides an intuitive single interface (via a laptop computer) for the crew to interact and participate in more advanced and realistic training scenarios. The astronauts would be able to train on an individual basis or remotely with SSTF in a seamless fashion.

In the operational realm, ground controllers should be able to easily visualize the current and future configurations of a space platform and be able to rapidly access critical information about the onboard systems. Access to repair manuals and problem reports, as well as the ability to easily generate and visualize station procedures (whether as part of training or responding to unplanned emergencies) are very useful assets not currently available in a unified framework system. The IVS proposes to provide this unified framework capability to revolutionize methods of accessing information and enhance operators' tools.

The final challenge to be listed here involves real-time data. In addition to the document data previously mentioned, real-time telemetry also constitutes information required by engineers and scientists. It is often not easily accessible, and even when it is, the sheer volume and complexity of the data can make analysis difficult. In the demanding environment of human spaceflight, it is perhaps not surprising that intense interest is focused on advanced techniques for data analysis. Analysis is done to both predict future events and to more completely understand current and past ones. Desirable solutions are ones that meet the requirements in a highly automated fashion while simultaneously adapting to changing conditions. Examples of problems in this category include data validation (determining if there are sensor failures in a collection of sensors) and health monitoring (using the current and past state of the system to predict future subsystem failures). A goal of the IVS is to build a framework that is useful for these kinds of intelligent tools and algorithms.

The next sections describe the research and software development being performed to address these challenges.

2. Approach

The Smart Systems Research Laboratory at the NASA Ames Research Center has developed an IVS prototype targeted for virtual training and operations for the ISS. The IVS architecture consists of three core subsystems: information management, visualization, and support for simulation and decision making tools.

Information management refers to providing pertinent data in a rapid fashion to both users and integrated software tools [2]. It encompasses both document data (e.g., design review documents, CAD files, and operational manuals) and real-time data (e.g., state information, audio, and video). The goal is to retrieve the data as necessary from multiple heterogeneous databases containing ISS information. An earlier publication described various aspects of this component of the IVS [3].

Visualization includes the use of two- and three-dimensional graphics to present information to the users, sometimes taking the form of virtual environments (VEs) [4]. The geometry of the VE is derived directly from ISS CAD files. Clicking on objects in the VE retrieves data associated with the object and presents the users with a hierarchy of the data.

Simulation and decision making support include among other things software designed to examine and analyze cross-systems effects and optimize training scenarios. A critical component of this architecture is providing useful APIs to the various subsystems, thereby allowing others to adapt the IVS framework to their needs. A decision-making capability currently available in the IVS deals with the ability to generate and visualize procedures in real-time. Users can generate, save, share, and playback procedures while navigating through the virtual environment. The IVS provides playback flexibility where the user can obtain any desired view.

Figure 2 illustrates some of these features. The center pane displays the interior of a module on the ISS known as the Centrifuge Accommodation Module. Selecting on an item in the component explorer highlights the physical component in the VE. That action in turn brings up associated data pertaining to the object in the document explorer. For the case in Figure 2, this is illustrated by clicking on the highlighted object known as the Life Sciences Glovebox.

3. Implementation

The Intelligent Virtual Station is coded in C++ using Microsoft Visual Studio .NET. Because of its heavy use of 3D graphics, C++ continues to afford advantages compared to Java when it comes to 3D rendering. The IVS has been designed and tested to run on Microsoft Windows platforms (2000/XP).

The database access technology for the information management core is ODBC. ODBC is well established and allows for straightforward access to nearly all commercial and open source relational databases [5].

Microsoft Foundation Classes (MFC) are used for the user interface [6]. OpenGL is used for rendering the virtual environment [7]. Previous versions of the rendering subsystem used proprietary graphic libraries and have been abandoned in favor of OpenGL for reasons of performance and licensing.

Figure 2: Screen capture illustrating components of the IVS.

The client portions of the IVS can run on modern laptop computers. The minimum platform IVS has been tested on is an Intel Pentium III 1.0GHz processor, 256MB of RAM, and an NVIDIA GeForce2Go with 32MB of VRAM.

4. Results

At the time of writing, the IVS has been extended to include interfaces to the Space Station Training Facility at the NASA Johnson Space Center. Given appropriate assets and network authorization, a user can navigate within the VE and select on an object (such as a caution and warning panel or an onboard laptop computer) and bring up a 2D interface into the SSTF. The interface allows the user to interact with the SSTF from local or remote locations.

The information management system currently can connect to a local database or a remote database. These documents include Word, PDF, Excel files; JPEG and GIF images; and AVI and MPG clips. These documents were not managed in their original sources but instead in single databases for use by the IVS.

The virtual environment currently consists of only five of the modules for the ISS: Centrifuge Accommodation Module (CAM), Node 2, Columbus, Japanese Experiment Module (JEM), and US Lab. Extensive internal coverage has been imported from CAD for Node 2 and the CAM, limited coverage for the US Lab, and sparse for Columbus and JEM.

5. Future Work

The current instantiation of IVS has limitations that future work will look to address and resolve. The information management system currently only addresses connections to individual databases. Developing, integrating, and utilizing federated database technologies will be required to access the disparate sources of data. Intelligent integration of telemetry sources (e.g., real-time and logged data) is a near-term goal. To enable multiple user collaborations from remote locations, the IVS will be expanded into a distributive collaborative virtual environment. As well, more work will be done to define the services and interfaces required by intelligent data validation and health monitoring tools and algorithms.

6. Conclusion

Several IT disciplines are involved in enhancing the state-of-the-art in virtual training and operations. These include areas such as information management technologies, collaborative virtual environments (VEs), and simulation and decision making tools. This paper described a software framework and prototype application known as the Intelligent Virtual Station, the goal of which is to support the training and operational needs of a wide variety of manned and unmanned space missions. Prototype software development is done to assess, validate, and obtain feedback on the architectural design efforts.

Although the IVS has been initially applied in the context of the ISS, this work is more broadly applicable to complex systems in general. The IVS design, architecture, and implementation are not ISS-specific. For example, any VE that has associated with it large amounts of data can benefit from a system such as the IVS. Potential space applications include spacecrafts, rovers, satellites, and other future space platforms. Foreseeable terrestrial applications include commercial airplanes, aircraft carriers, cars, and tanks.

The current version of the Intelligent Virtual Station prototype application was deployed at the NASA Johnson Space Center in the fourth quarter of 2002 with groups that work with the International Space Station. Future deployments at a larger scale and at other NASA centers are scheduled to follow in 2003.

References

[1] NASA Human Spaceflight, International Space Station, http://spaceflight.nasa.gov/station/ (current Feb. 2003).

[2] U. Nambiar et al., "Current Approaches to XML Management", *IEEE Internet Computing,* 6(4), 2002, pp. 43-51.

[3] 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.

[4] K.M. Stanney, ed., *Handbook of Virtual Environments,* Lawrence Erlbaum Assoc., Mahway, NJ, 2002.

[5] K. Geiger, *Inside ODBC,* Microsoft Press, Redmond, WA, 1995.

[6] E. Olafsen, K. Scribner, and K.D. White, *MFC Programming With Visual C++ 6 Unleashed,* SAMS Publishing, Indianapolis, IN, 1999.

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