

A DATA MANAGEMENT SYSTEM FOR INTERNATIONAL SPACE STATION SIMULATION TOOLS

Bradley J. Betts, Rommel Del Mundo, Sharif Elcott, Dawn McIntosh, Brian Niehaus, Richard Papasin, Robert W. Mah

Smart Systems Research Laboratory
NASA Ames Research Center, M/S 269-1
Moffett Field, CA 94035
United States of America

ABSTRACT

Groups associated with the design, operational, and training aspects of the International Space Station make extensive use of modeling and simulation tools. Users of these tools often need to access and manipulate large quantities of data associated with the station, ranging from design documents to wiring diagrams. Retrieving and manipulating this data directly within the simulation and modeling environment can provide substantial benefit to users. An approach for providing these kinds of data management services, including a database schema and class structure, is presented. Implementation details are also provided as a data management system is integrated into the Intelligent Virtual Station, a modeling and simulation tool developed by the NASA Ames Smart Systems Research Laboratory. One use of the Intelligent Virtual Station is generating station-related training procedures in a virtual environment. The data management component allows users to quickly and easily retrieve information related to objects on the station, enhancing their ability to generate accurate procedures. Users can associate new information with objects and have that information stored in a database.

KEY WORDS

data management system, virtual environment

1. INTRODUCTION

The International Space Station (ISS) is a collaborative engineering and scientific endeavor among sixteen nations. Its design and ongoing assembly and operation are complex undertakings. Given this complexity, it is not surprising that design, training, and operational demands of the ISS require extensive support from modeling and simulation tools. Equally unsurprising is that the ISS has associated with it large amounts of electronic data, such as design documents, CAD files, and electrical wiring diagrams, to list but a few.

Through work developing modeling and simulation tools for the ISS, a critical need has become apparent: *to provide access to large amounts of ISS data, and to do so directly within the simulation environment.*

Although sometimes ancillary to the actual execution of a particular simulation, ISS data can have a significant impact on correctly interpreting simulation results and in deciding what simulation to conduct in the first place. Making this data accessible within the simulation environment saves users from having to work outside the tool (possibly using a variety of applications to access data, such as database access applications and web-based information retrieval) and can increase the probability that users have timely access to the information they need.

Many different teams that deal with the ISS, ranging from training to operations to engineering design teams, have identified the challenge of needing access to large volumes of ISS data within simulation and modeling tools. Others have identified similar challenges in contexts other than the ISS [1], [2], [3], [4]. Research conducted by the Smart Systems Research Laboratory at the NASA Ames Research Center has and continues to address this challenge, in part by looking at ways to integrate *data management systems* directly into modeling and simulation environments.

At its most basic, a data management system must provide users the ability to store, retrieve, and manipulate data [5]. This paper seeks to demonstrate that providing data management services within a simulation environment can provide benefit to users. By discussing a specific implementation it is shown that so doing need not be difficult—a data management system has been integrated into a simulation tool developed at the Smart Systems Research Laboratory. The tool is known as the Intelligent Virtual Station. A key component of this simulation environment is a virtual reality model of portions of the ISS (information on recent advances in virtual environments can be found in [6]). The data

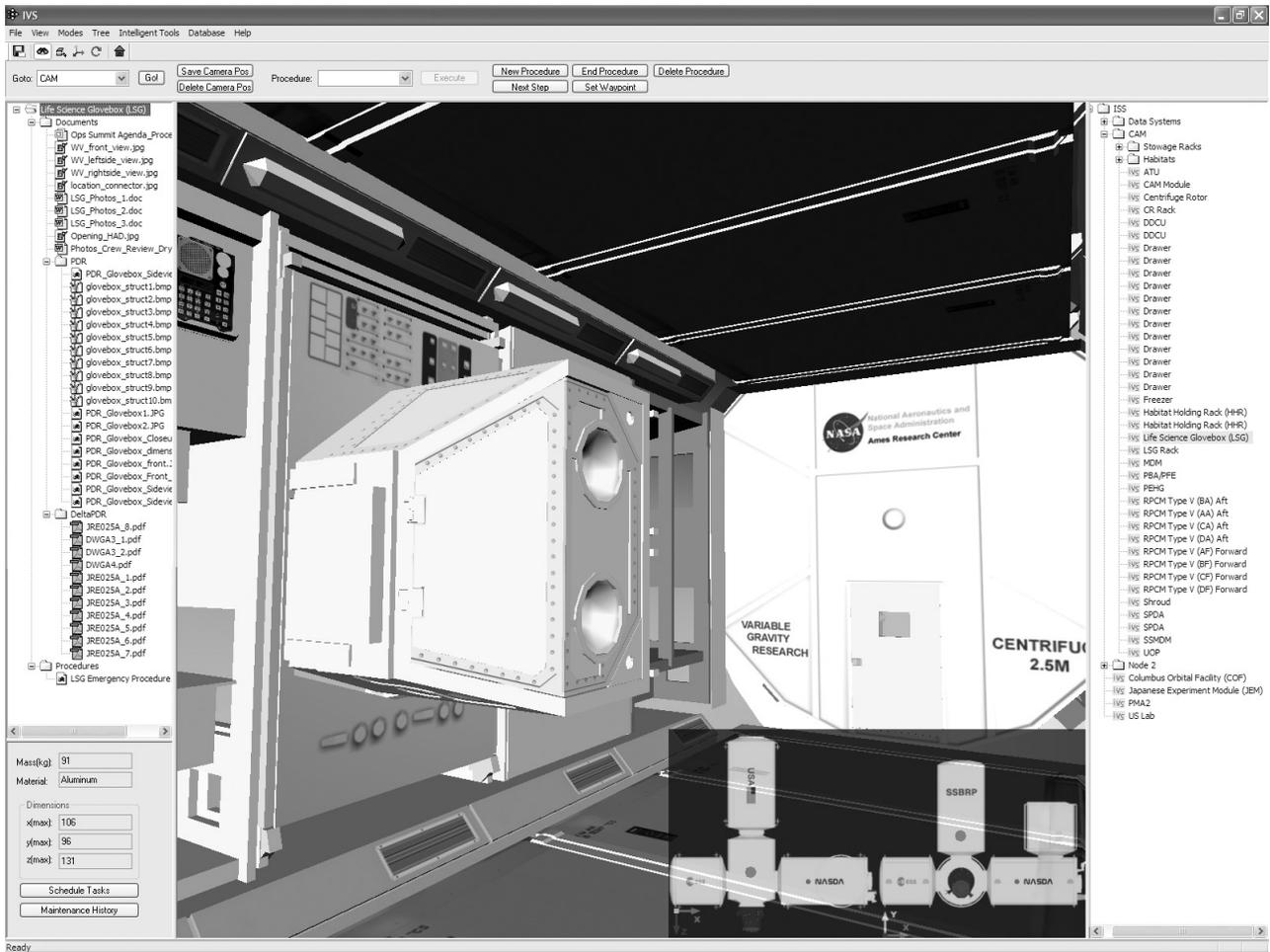


Figure 1. Screen capture of Intelligent Virtual Station application.

management system provides a link between objects in the virtual environment and data related to that object. Users have access to relevant data by simply clicking on an object in the virtual environment.

The remainder of this paper indicates the approach taken to integrating a data management system into a modeling and simulation environment. Implementation details and results for the data management system developed for use in the Intelligent Virtual Station are provided. Finally, conclusions and avenues for future work are presented.

2. APPROACH

The aforementioned Intelligent Virtual Station is a modeling and simulation tool with a variety of different uses. For example, completion of work in progress will see the tool used to generate training procedures for astronauts, to help them visualize the steps required to replace a component on the station. Figure 1 shows a screen capture of the application. In the center panel is a

rendering of the viewable portion of the virtual environment, in this case the interior of a module on the ISS known as the Centrifuge Accommodation Module. Also shown in this center panel, in the bottom right corner, is a location map to help users understand where they are located in the station. The Intelligent Virtual Station application runs on Microsoft Windows machines using standard displays, i.e., it is a non-immersive virtual environment. Users can navigate through the virtual environment using a standard keyboard and mouse. Extensions to more advanced interfaces are possible but not a focus of this effort.

This paper is focused on the data management component of the Intelligent Virtual Station. In designing such a system, an early and important question is to decide exactly what data is to be managed. Ideally any and all data associated with the station—including real-time monitoring and control information—would be made available to users of the Intelligent Virtual Station, subject to authorization settings for a given user (for more on authorization in data management systems see [7]). Users should have the ability to create, move, and delete information.

In the case of the ISS, a reasonable place to start is to provide data management capabilities for the vast amount of digital document data that exists for the station. Although by no means in all cases, oftentimes documents are associated with individual objects or groups of related objects on the station. For example, a device will likely have a variety of design review documents associated with it, as well as an operations manual. The approach taken was to provide users with the ability to associate a hierarchical tree of documents with individual objects in the visualization. This association can be seen in Figure 1. The panel on the right shows the set of selectable objects in the scene (themselves organized into a hierarchy) and is referred to as the model explorer. An object has been selected, in this case the Life Science Glovebox, and its associated tree of arbitrary document data is automatically shown in the document explorer panel on the left.

From the user point of view, the document explorer acts as a conventional file explorer: documents can be added, moved, retrieved, and deleted. In fact the document explorer is a view into a database, not the local file system running the Intelligent Virtual Station application. The hierarchy of selectable objects shown in the model explorer is defined in the original CAD files used to derive the virtual reality models (the model format used is a binary derivative of VRML97 [8]).

This kind of association—objects in the virtual environment linked to their document data—allows users of the Intelligent Virtual Station to rapidly retrieve information useful when performing simulations. An example is replacement-procedure generation. Many devices on the ISS are designed to be replaced by the astronauts in the event of failure or impending useful lifetime expiration. These devices, known as Orbital Replacement Units, have detailed text-based procedural documents associated with them, precisely outlining the sequence of steps required to replace them (sometimes an animation of the steps is also provided by the device designer). The data management system allows procedural documents to be associated with their Orbital Replacement Units in the virtual environment. By so doing, the document is provided with a spatial context for interpretation and is easily and naturally retrievable.

3. IMPLEMENTATION

The data management system (and indeed the entire Intelligent Virtual Station) was coded in C++ using Microsoft Visual Studio .NET to run on Microsoft Windows platforms. As a graphics-intensive application, C++ continues to afford advantages compared to Java when it comes to rendering 3D graphics. Microsoft Foundation Classes are used for the user interface and OpenGL is used for rendering the virtual environment (earlier versions of the rendering subsystem used proprietary graphics libraries that were abandoned in

favor of a pure OpenGL approach for reasons of performance and licensing).

The database access technology used to store the document data is ODBC [9]. Although Microsoft provides several other database access technologies (e.g., OLE DB, DAO, ADO.NET), ODBC is well established and allows for straightforward access to nearly all commercial and open source relational databases. As well, its similarity to JDBC will allow for easier porting of the data management system to Java should that be required in future iterations (e.g., to deploy on a wider array of operating systems).

The data management system at present defines its own database schema. This leads to potential duplication if the document resides in another ISS database. That in turn leads to versioning issues. While the current implementation of the data management system has proven useful, future iterations will see in-place management of document data, eliminating the duplication issue and alleviating the versioning problem. This longer-term goal of in-place management of a variety of heterogeneous databases influenced the decision to use relational database features (as opposed to object-oriented features). It may prove neither useful nor feasible to port existing relational ISS databases to object-oriented ones.

A claim of this paper is that providing a data management system in a modeling and simulation environment need not be difficult. Figure 2 shows a simple database schema that can be used to associate a hierarchical tree of data with objects in a virtual environment. This schema has been implemented in the data management system of the Intelligent Virtual Station.

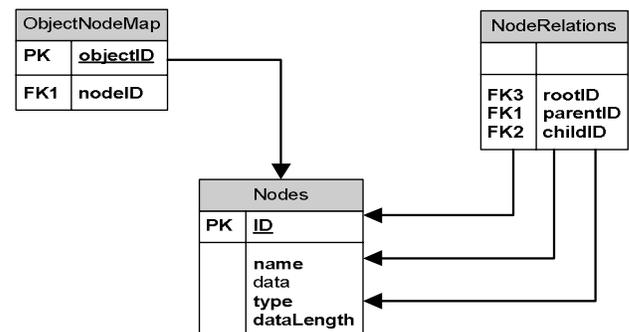


Figure 2. An entity-relationship diagram for document trees. Required elements are shown in bold font, primary keys are underlined.

Each item in the document tree is referred to as a *Node*. A Node can be either non-terminal (i.e., a folder) or terminal (i.e., a file); terminal nodes have document data associated with them. Parent-child relationships are

captured in the *NodeRelations* table. Providing a root identifier in this table schema is redundant and done only to facilitate querying. So as to be applicable to as wide a variety of relational databases as possible, database-specific SQL extensions are deliberately not taken advantage of (such as Oracle's "CONNECT BY" clause). Notice that this schema allows the same document to appear in multiple trees if so desired (a hard-linking option). For example, *NodeRelations*(1,1,3) and *NodeRelations*(2,2,3) would have Node 3 appear as a child of both Node 1 and Node 2 (which in turn are root nodes in this example). The table *ObjectNodeMap* provides the association between selectable objects in the virtual environment and the root node of a document tree.

Actually making use of the schema in Figure 2 is straightforward. Figure 3 shows a UML [10], [11] diagram of a collection of classes that do just that (some ancillary classes, such as the entire exception hierarchy, have been omitted for clarity, as have methods and members). Once a *DocumentTree* object is actually created, displaying it to the user (i.e., the document explorer pane on the left of Figure 1) is done by making use of the *CTreeCtrl* Microsoft Foundation Class [12]. When implemented in C++, the entire set of classes was implemented using just over three thousand lines of code.

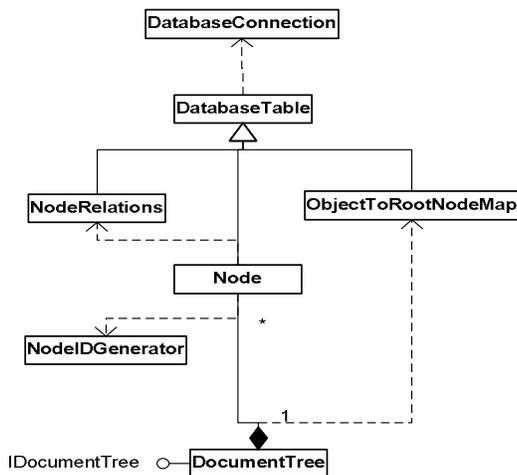


Figure 3. A UML diagram showing the principal classes involved in the implementation of the document tree. Ancillary classes, methods, and members have been omitted.

The relative simplicity of the schema outlined in Figure 2 leads to good scalability properties. The time required to construct a document tree grows linearly with the number of elements in the tree.

4. RESULTS

At the time of writing, the data management system of the Intelligent Virtual Station had 407 different ISS documents under management (with a corresponding database size of 317 megabytes). These documents include Word files, PDF files, Excel files, JPEG and GIF images, and AVI clips. These documents were not managed in their original databases but instead in a single database created for use by the Intelligent Virtual Station.

The data management system has had a minimal impact of the runtime performance of the Intelligent Virtual Station. This is of particular concern given the need to render the virtual environment in real-time. The typical platform specification upon which the Intelligent Virtual Station is run at present is an Intel Pentium 4 1.2 GHz processor, 256 megabytes of RAM, and a NVIDIA GeForce 2 with 32 megabytes of RAM.

5. CONCLUSIONS

A data management system has been added to a modeling and simulation environment, providing a link between objects in a virtual environment and a hierarchical tree of document data. An implementation approach has been developed and shown to be straightforward while at the same time conveying benefits to users—information can be easily retrieved, with spatial context, and applied to facilitate the modeling process.

Although presented in the context of the International Space Station, this work is more broadly applicable. For example, any virtual environment that has associated with it large quantities of reference document data can benefit by adopting a data management system as outlined in this paper.

The Intelligent Virtual Station application, including its data management system, will see limited deployment with groups that work with the International Space Station in the fourth quarter of 2002. Larger-scale deployments will follow in 2003.

The approach outlined in this paper has limitations that future work will look to address. The data management system outlined here associates data with specific objects or groups of objects in the virtual environment. Not all data, however, are amenable to hierarchical organization or to association with specific objects (e.g., real-time attitude information for the International Space Station). Another limitation is that the database schema presented here does not provide a version control solution to the documents placed in it. Related to this issue, the approach taken here calls for potential duplication of existing sources of data instead of providing in-place management of (possibly) heterogeneous databases. Finally, explicit provision has

not been made for searching through documents managed by the system.

ACKNOWLEDGEMENTS

The authors wish to thank Dr. Edward Wilson of Intellization for his insight and advice, as well as for his help preparing this paper. The authors also wish to thank Michael Guerrero of the NASA Ames Research Center for his contributions to the virtual reality models used in the Intelligent Virtual Station.

REFERENCES

- [1] D. A. Reed, R. C. Giles, & C. E. Catlett, Distributed data and immersive collaboration, *Communications of the ACM*, 40(11), 1997, 39-48.
- [2] Z. Lacroix, Biological data integration: wrapping data and tools, *IEEE Trans. on Inform. Tech. in Biomedicine*, 6(2), 2002, 123-128.
- [3] C. A. Murphy & T. D. Perera, The definition and potential role of simulation within an aerospace company, *Proc. 2001 Winter Simulation Conf.*, Arlington, VA, 2001, 829-837.
- [4] J. P. Shewchuk, K. H. Chung, & R. C. Williges, Virtual environments in manufacturing, in K. M. Stanney (Ed.), *Handbook of virtual environments* (Mahway, NJ: Lawrence Erlbaum Assoc., 2002), 1119-1141.
- [5] U. Nambiar, Z. Lacroix, S. Bressan, L. Mong, & L. Yingguang, Current approaches to XML management, *IEEE Internet Computing*, 6(4), 2002, 43-51.
- [6] K. M. Stanney (Ed.), *Handbook of virtual environments* (Mahway, NJ: Lawrence Erlbaum Assoc., 2002).
- [7] D. Raymond, Authorization in data management systems, *Proc. IEEE Conf. on Security Tech.*, London, UK, 2001, 202-212.
- [8] A. L. Ames, D. R. Nadeau, & J. L. Moreland, *VRML 2.0 sourcebook, 2nd ed.* (New York: John Wiley & Sons, 1997).
- [9] K. Geiger, *Inside ODBC* (Redmond, WA: Microsoft Press, 1995).
- [10] J. Rumbaugh, I. Jacobson, & G. Booch, *The unified modeling language reference manual* (Reading, MA: Addison-Wesley, 1999).
- [11] G. Booch, J. Rumbaugh, & I. Jacobson, *The unified modeling language user guide* (Reading, MA: Addison-Wesley, 1999).
- [12] E. Olafsen, K. Scribner, & K. D. White, *MFC programming with Visual C++ 6 unleashed* (Indianapolis, IN: SAMS Pub., 1999).