Dshell++: A Component Based, Reusable Space System Simulation Framework

Christopher S. Lim and Abhinandan Jain,
Jet Propulsion Laboratory California Institute of Technology 4800 Oak Grove Drive, Pasadena, CA 91109

***Abstract*—Traditionally, spacecraft simulations have been built as a monolithic programs targeted to a specific application. Adding or changing a feature or reusing components required a deep understanding of the entire simulation. Dshell++ is a high fidelity, multi-mission, physics-based simulation toolkit with the goal of increasing simulation reuse productivity by using modern object-oriented techniques to allow component reuse, a data-driven architecture, and a dynamic run-time interface that for complex, high-fidelity spacecraft simulations.**

***Index Terms*—Simulation Software**

I. Introduction

Dshell++ is a high fidelity, multi-mission, physics-based simulation toolkit with the goal of increasing simulation reuse productivity by using modern object-oriented techniques to allow component reuse, a data-driven architecture, and a dynamic run-time interface that for complex, high-fidelity spacecraft simulations. Dshell++ uses object-oriented techniques to allow code reuse and component building. Dshell++ simulations consist of a collection of component device models from model libraries assembled and connected together into a data flow to meet the required simulation behavior. For example, a thruster model is derived from an actuator object (a device that applies a force on a body) and inherits alls the interfaces and properties of the actuator object. Dshell++ provides facilities for the inter-connection of and efficient data exchange between such models. Related models within sub-systems (for example, a bank of thrusters) can be grouped together into a "subassembly" which in turn can be part of larger subsystem. The assemblies are reusable and can be used across more than one simulation. This allows complex simulations to be built by simply choosing and connecting the desired components together.

Dshell++ is data-driven. Which models to include and how the models are connected and grouped together, along with the initial states and parameters, are entirely specified through configuration files which are processed at run-time. If a new, improved model becomes available, incorporating the new model is straightforward - simply swap out the old model with the new one in the configuration file with no recompilation required.

Dshell++ includes auto-generated Python bindings to its C++ classes. The Python scripting language interface to the Dshell++ classes allows users to access and even extend the simulation functions and classes at run-time entirely in Python without modifying the C++ classes. Python provides all of the extensive features of a modern software language as well as additional ones including parsers, run-time loading of extensions, large collection of open source Python modules that are available to the simulation developers and users. Watch functions can be created to trigger on events and monitor or plot data on your screen. Visualization in real-time (such as watching a rover slipping down a slope) is possible through the Dspace 3D toolkit. Interfaces can be built to external applications (such as Matlab).

Dshell++ is portable. The underlying framework is written in standard C++ and is highly optimized to run in real-time on laptops, desktop workstations and supercomputers.

Dshell++ includes the multi-mission high-performance DARTS flexible multibody dynamics module based on the Spatial Operator Algebra framework for solving the dynamics of spacecraft multi-body dynamics. Dshell++ has been used to develop real-time simulations for cruise/orbiter vehicles as well as to develop domain specific simulators such as ROAMS for surface rover simulations and the DSENDS entry, descent and landing simulator. These simulators have been in use by a number of NASA missions (Mars Science Laboratory, Phoenix) as well as technology development activities (Athlete, Lunar Surface Operations Simulator).

References

[1] J. Biesiadecki, D. Henriquez, and A. Jain, "A Reusable Real-Time Spacecraft Dynamics Simulator," in *16th Digital Avionics Systems Conference*, (Irvine, CA), Oct. 1997.

[2] A. Jain and G. Rodriguez, "A Spatial Operator Algebra for Compuational Multibody Dynamics," in *International Conference on Scientific Computation and Differential Equations (SciCADE'97)*, (Grado, Italy), Sep. 1997.

[3] H. Nayar, et.al., "A Lunar Surface Operations Simulator," in *Proc. International Conference on Simulation, Modeling and Programming for Autonomous Robots (SIMPAR 2008),* (Venice, Italy), Nov. 2008.

Filename: 16.doc
Directory: D:\F Drive_new\www-smcit09\abstracts-contri_papers
Template: C:\Documents and Settings\klittle\Application Data\Microsoft\Templates\Normal.dot
Title: .
Subject: IEEE Transactions on Magnetics
Author: -
Keywords:
Comments:
Creation Date: 9/3/2008 9:34:00 AM
Change Number: 6
Last Saved On: 11/2/2008 4:38:00 PM
Last Saved By: home1
Total Editing Time: 35 Minutes
Last Printed On: 5/5/2009 2:01:00 PM
As of Last Complete Printing
Number of Pages: 1
Number of Words: 710 (approx.)
Number of Characters: 4,272 (approx.)