4th Responsive Space Conference
RS4-2006-4005

Responsive Space's Spacecraft Design Tool (SDT)

Robert Strunce, Fred Eckert,
Craig Eddy
Star Technologies Corporation
Great Falls, VA

4th Responsive Space Conference
April 24–27, 2006
Los Angeles, CA

RESPONSIVE SPACE'S
SPACECRAFT DESIGN TOOL (SDT)

Robert Strunce
Fred Eckert
Craig Eddy

Star Technologies Corporation
731 Walker Road, Suite G1
Great Falls, VA 22066

ABSTRACT[1]

Star Technologies Corporation has developed a ".Net Framework Simulation Architecture", "Spacecraft Design Tool" (SDT), which is an open framework for Responsive Space's rapid design concept spanning mission capture to deployment. SDT has been incorporated in all three test cells at AFRL's Responsive Space Testbed: FlatSat Integration Cell, Responsive Satellite Demonstration Cell, and Responsive Avionics Technology Cell. SDT is integrated into the Mission Design phase for spacecraft simulation and analysis as well as providing the real-time spacecraft dynamics/kinematics, earth environment, and sensor/actuator models in support of Hardware-In-Loop tests. SDT can simulate a sensor/actuator, or interface to its respective hardware emulator or to the actual hardware through the Responsive Space's Plug-n-Play electronic environment. SDT provides a true software PnP environment where the User can seamlessly inherit properties from within SDT as well as add his own component or subsystem capabilities such as complex propulsion or electrical power subsystems. SDT has a 3D Visual game engine for rendering 3D objects such as articulated spacecraft, earth, sun, and moon. SDT has been used to model TacSat2, and a generic Responsive Space satellite.

This paper describes SDT's .Net Framework Simulation Architecture that supports true software PnP; SDT's support of Mission Design and Analysis; SDT's real-time implementation and utilization in the three test cells in AFRL's Responsive Space Test Bed that supported both TacSat2 and TacSat3.

INTRODUCTION

This paper will discuss the implementation and utilization of SDT within the AFRL's Responsive Space Testbed, RST. AFRL's Responsive Space Testbed is comprised of three test cells (Figure 1): FlatSat Integration; Responsive Satellite; Avionics Technology.

The ***FlatSat Integration Cell*** is initially used to host a demonstration of the Joint Warfighter Space, D2 satellite, developed by MicroSat Systems Inc. The demo consists primarily of checkout procedures of the various avionics systems.

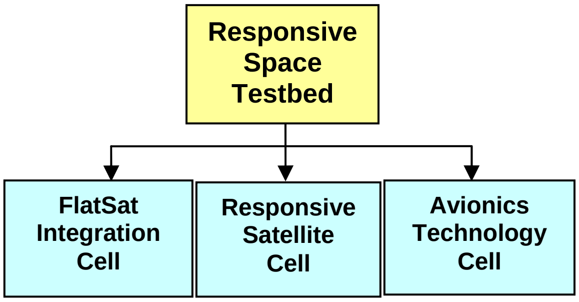

Figure 1. Responsive Space Testbed

The ***Responsive Satellite Cell*** is used to demonstrate the responsive avionics concepts integrated into a demonstrator vehicle and evaluate it from design, assembly, test and mission perspectives. The design tool pipeline is used to create the design in response to mission objectives, create build instructions, assist in the automated checkout procedures that run during assembly and create packaging information used by the flight computer and the ground system software. Day in the life mission scenarios can be performed in the Responsive Satellite Cell to explore the impact of improved methodologies.

Figure 2 shows how the Responsive Satellite Cell was configured to deliver the simulated environment to the Hardware-In-Loop (HWIL) Device Under Test (DUT). SDT provides the Detailed Simulation and the Spacecraft Drag & Drop functions, as well as host the Graphical User Interface (GUI) Data Displays.

The ***Responsive Avionics Technology Cell*** is used to integrate and test core responsive avionics technologies including Plug 'n Play avionics, Software Defined Radio, Adaptive Wiring Manifold, High Performance Computing On Orbit (HPCOO) and the Satellite Data Model (SDM). It is intended to cover a broad range of technologies in proof of concept, experimental state of maturity.

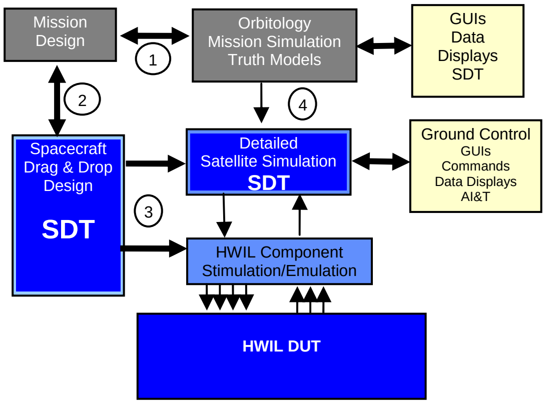
Mission Design
1
Orbitology Mission Simulation Truth Models
GUIs Data Displays SDT
2
4
Spacecraft Drag & Drop Design
SDT
Detailed Satellite Simulation
SDT
Ground Control GUIs Commands Data Displays AI&T
3
HWIL Component Stimulation/Emulation
HWIL DUT

Figure 2. RST Functional Partitions

Star Technologies' Spacecraft Design Tool is used in all three test cells in several capacities. The common function that it provides is a six degree of freedom simulation of the spacecraft and its environment. It also provides the simulation information in real time to the Hardware-In-Loop system where it is tailored to device-specific characteristics prior to exchange with the flight computer(s). Finally, the SDT simulation framework was used to host and integrate the Test Conductor Console for the Flat Sat and Responsive Sat cells.

SDT SIMULATION FRAMEWORK

Star Technologies has developed a powerful software architecture to build component-based simulations with Microsoft's .Net Framework which provides the basic development platform as , depicted in Figure 3. Atop those core technologies, Star Technologies has developed a Component based simulation framework that is used across all of its products. Finally, the Spacecraft Design Tool utilizes features in both Frameworks to deliver a state of the art software tool.

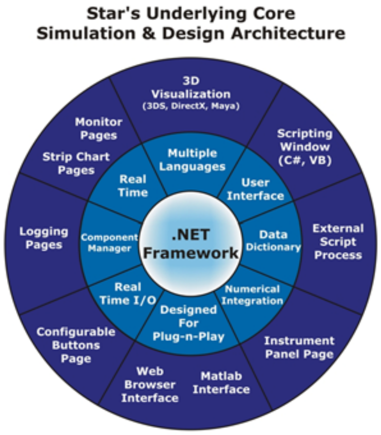

Figure 3. SDT Framework

Several key technologies in .Net are exploited in SDT to provide a powerful, extensible feature set. Nearly 40 languages are available to use for implementation and / or scripting. SDT currently hosts C#, Basic, Jscript and XML as scripting languages. Scripts are compiled prior to execution, so there is no performance penalty. The .Net Code DOM compiler is invoked for a script after editing in the running simulation, allowing very rapid customization. The .Net Fusion loader allows all component models to be loaded on demand, as required by the scenario. A plug-in manager knows how to find required modules.

Even though Windows is a virtual Operating System and can page fault at the most inopportune time, the SDT Framework creates all components during initialization that are required in the real time threads. Displays that need dynamic memory are allowed to page fault, since they run in the user interface thread. Real time I/O is provided with dedicated threads per I/O channel. All real time threads can have their priority set above the Windows real time threshold, insuring good response.

The SDT Framework incorporates an active Data Dictionary that can be modified while the simulation is active. It provides a hierarchical namespace for most objects contained by the running components. Data Dictionary Data Items can be dragged from the Data Dictionary tree view to other controls and displays available within SDT to add that object to the display. You can drag a Data Item to a Monitor view to see text values, a strip chart, Mercator view and 3D visualization. Those Data Items can even be the spacecraft object. Additionally, Data Items may be logged to a file or stream, scatter/gathered to and from the real time I/O messaging, used in graphical instrument panels with controls and displays, exchanged with Matlab and even onto a web browser.

The SDT Framework uses a Component as its fundamental unit of configurability and work. Components are loaded from a prioritized list of library directories and initialized to the characteristics in the component characteristic file. Characteristics may be specified at the part or serial number level. Vehicles are built up from a hierarchical list of components and multiple vehicles and environmental models may be loaded. Scenario initialization data may uniquely configure a component's initial conditions each time the simulation is initialized. Components may be scheduled based on time period or from the State Space integration algorithm.

SPACECRAFT DESIGN TOOL

Star Technologies' Spacecraft Design Tool is constructed from the two Frameworks previously mentioned. It is a .Net application that loads a scenario as its primary runnable document. The scenario specifies a component configuration, an initialization script and data, visible views and the I/O configuration. A Block Diagram Editor is used to edit the hierarchical component configuration using libraries of components. A specific set of spacecraft components includes environmental, 6DoF dynamics, visual, sensor, actuator, electrical and propulsion building blocks. The ADCS software may be run in exactly the same form as it has in the flight computer, utilizing stub interfaces into the Data Dictionary to mimic the flight computer's I/O API. See Figure 4.

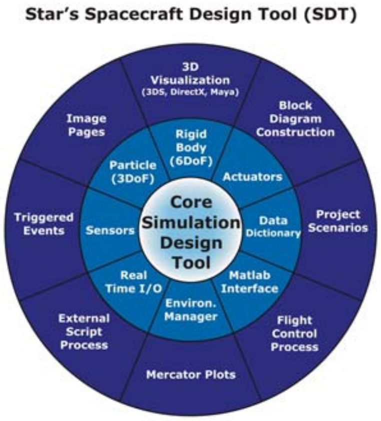

Figure 4. Spacecraft Design Tool

Two views that need further discussion are the 3D visualization and an image view. The 3D visualization defaults to an ECI based view of the Earth, with any other environmental components displayed as required. Any number of spacecraft in the scenario may be viewed in multiple 3D view ports. The visual representation can come from many standard 3d formats, including; 3ds, mb, X and SLDASM. The camera may be assigned to any visible component in the view and referenced to any of the coordinate frames associated with that object. Most spacecraft present ECI, ECF, Body and LVLH coordinate frames. Three dimensional annotations may be added to any object, including: vectors, lines, planes, triads (coordinate frames), spherical reticules, fields of view/regard and orbits/ground tracks. Data Items may be displayed on a 2D overlay of the visualization.

The image view may be tied to any location on the earth and/or any sensor field of view, say for instance the pierce point of a space telescope. It may be zoomed, panned, rotated and shown as a field of view within a field of regard. A 10 meter imagery of China Lake was used for demonstration purposes. Star Technologies plans to purchase global 15 meter imagery to provide detailed information and customers may add their own imagery.

Finally, there is a triggered event and timed variable timeline that allows users to drive simulation events and variables in order to mimic external commands.

Figure 5 shows the standard SDT application running a TacSat mission.

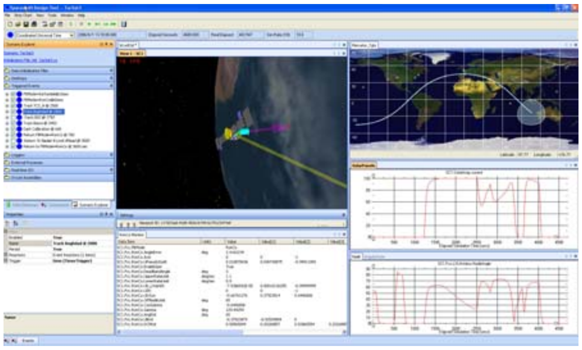

Figure 5. Spacecraft Design Tool

RESPONSIVE SPACE TECHNOLOGY

SDT was installed on three workstations at the RST lab at the AFRL, each one in a different cell. Additionally, Design-Net Engineering[1] installed their Configuration Solver on each workstation. They also added a Test Conductor's Console, electrical models, Responsive Sat components and custom views to Star Technologies SDT Framework. They leveraged the SDT framework by building their additional Components as SDT plug-in's, achieving high levels of integration and performance. The simulation workstation, shown in Figure 6, consists of three 50 inch HD Plasma displays, two 24 inch HD LCDs and a Dell Xeon workstation running Windows 64. Figure 6 is taken from the Responsive Satellite Cell.

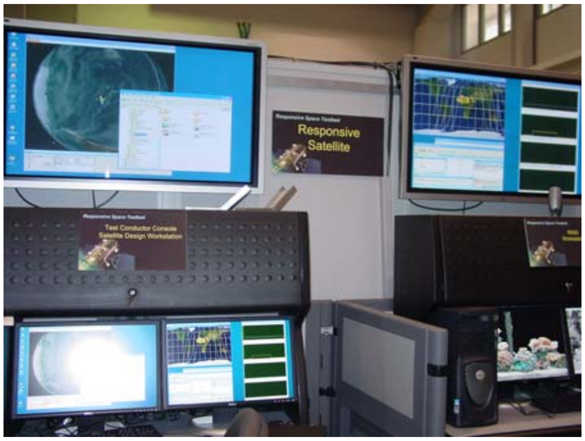

Figure 6. RST Workstation

SDT is used to drive all of the views on the displays, including the 3D visualization, Mercator map, electrical state display, Test Conductor Console and image views. A 100 Mbs Ethernet is used to connect SDT with the HWIL system provided by Applied Solutions Inc. A common 50ms frame period was used between SDT and the HWIL systems.

The Responsive Satellite shown in Figure 7 is a technology demonstrator which is located behind the workstation shown in Figure 6. The structure, provided by Spaceworks, is hogged out, anodized aluminum. Each panel of the six sided rectangular prism is attached to the other using 2 or 3 hinges. The external components are purely mocked up, except for the Appliqué Sensor Interface Module (ASIM), developed by SAIC.

Each sensor and actuator ASIM was connected via a Test Bypass Port to the HWIL system, allowing the simulation to provide the component environment for the flight computer, located inside the bus. This is depicted in Figure 7. The Responsive Sat, as shown, was assembled in a little over 2 hours, using a build list created from the Configuration Solver tool.

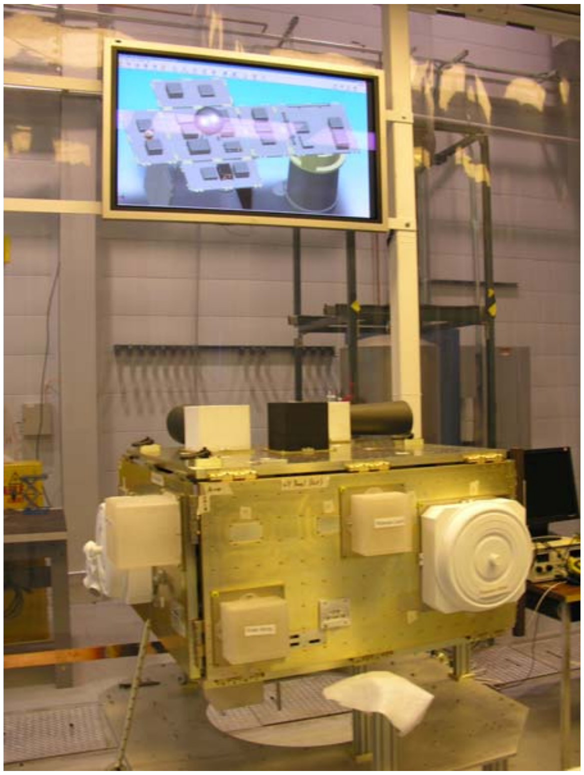

Figure 7. Responsive Satellite

All of the light blue symbols in Figure 8 depict SDT component models. The yellow symbols show Test Bypass Ports driven from the HWIL system that supply the dark blue ASIMs with real time data. Each ASIM is actually plugged into its Robust Hub using the USB bus, which eventually gets to the flight computer and the Satellite Data Model's Sensor Manager. It is the combination of USB bus, Electronic Data Sheets (xTEDS) description and Sensor Manager that makes Plug & Play Avionics really work. Each time a sensor is plugged into its Robust Hub, the Sensor Manager recognizes it, asks for its xTEDS and queues the RIMS ground station to begin automatic self test. The running processes on the flight computer interface through the Data Manager to Sensor Manager I/O.

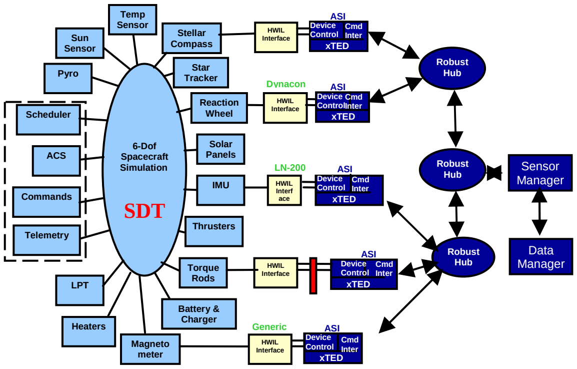

Figure 8. Hardware In Loop Interconnect

MISSION DESIGN AND ANALYSIS

The JWS D2 satellite, scheduled for the Flat Sat Cell, was still undergoing integration and test as of Initial Operational Capability (IOC) on December 6, 2005. However, its visualization, shown in Figure 9, was used in the Flat Sat Cell for mission design and analysis.

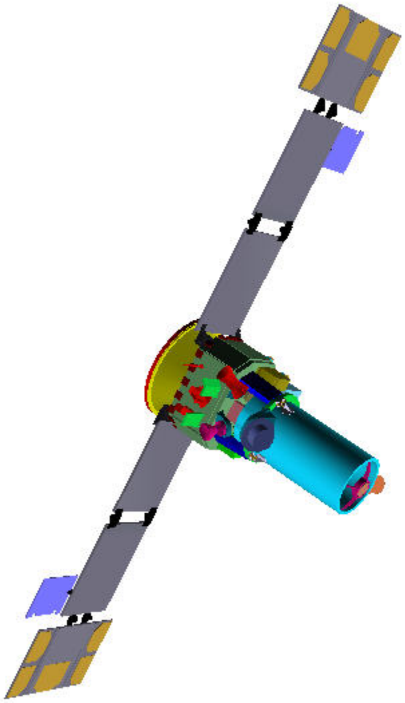

Figure 9. Roadrunner, JWSD2

The Configuration Solver interviews the user (designer) for mission parameters and continuously winnows the component library down to a build-able vehicle. The spacecraft configuration is then used to configure the simulation configuration as well as packaging information for the flight computer and automation information for the RIMS-based ground system, provided by Interface Control Systems.

One of the custom views that Design-Net Engineering created for RST depicts the state of the electrical system, specifically battery state of charge, solar array output and system load as shown in Figure 10. The display is driven asynchronously from electrical component state change events. You can see active reaction wheel loads in the lower chart.

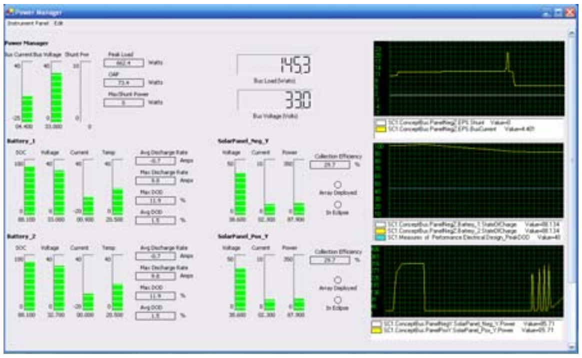

Figure 10. Electrical State

This capability was created as an extension to Star Technologies' Spacecraft Design Tool taking advantage of the SDT API and the underlying Microsoft .Net Framework. The electrical components monitor power output from the battery and solar panels and the power draw from all enabled components to correctly simulate a mission ensuring a power-positive state during spacecraft operations. In addition to the electrical components, TriSept Corporation, a subcontractor to Design-Net, provided an intuitive power display panel using a combination of SDT-supplied and custom extended graphical instruments to visually demonstrate the electrical power status of the mission during the simulation.

ACKNOWLEDGEMENTS

Thanks to James C. Lyke Civ AFRL/VSSE for his vision of Plug-n-Play electronic components and his drive to create the Responsive Space Testbed. Thanks to Don Fronterhouse for keeping the focus on the Responsive Space Testbed and herding all the cats to make it come together for IOC.

REFERENCES

1. K. Center, G. Murphy, R. Strunce, “Software as a Tall Pole in Achieving Rapid Configuration and Integration“, *AIAA Third Responsive Space Conference*, 2005.