



Software

Eyes on the Earth 3D

Eyes on the Earth 3D software gives scientists, and the general public, a real-time, 3D interactive means of accurately viewing the real-time locations, speed, and values of recently collected data from several of NASA's Earth Observing Satellites using a standard Web browser (climate.nasa.gov/eyes). Anyone with Web access can use this software to see where the NASA fleet of these satellites is now, or where they will be up to a year in the future. The software also displays several Earth Science Data sets that have been collected on a daily basis. This application uses a third-party, 3D, real-time, interactive game engine called Unity 3D to visualize the satellites and is accessible from a Web browser.

This work was done by Anton I. Kulikov, Paul R. Dornilla, Viet T. Nguyen, Randal K. Jackson, William M. Greene, Kevin J. Hussey, Christopher M. Garcia, and Christian A. Lopez of Caltech; Justin M. Moore and Andrea Boeck of Mooreboeck, Inc.; and Kevin Lane of Bohemian Grey for NASA's Jet Propulsion Laboratory. For more information, contact iaoffice@jpl.nasa.gov.

This software is available for commercial licensing. Please contact Dan Broderick at Daniel.F.Broderick@jpl.nasa.gov. Refer to NPO-47782.

Target Trailing With Safe Navigation for Maritime Autonomous Surface Vehicles

This software implements a motion-planning module for a maritime autonomous surface vehicle (ASV). The module trails a given target while also avoiding static and dynamic surface hazards. When surface hazards are other moving boats, the motion planner must apply International Regulations for Avoiding Collisions at Sea (COLREGS). A key subset of these rules has been implemented in the software. In case contact with the target is lost, the software can receive and follow a "reacquisition route," provided by a complementary system, until the target is reacquired. The programmatic intention is that the trailed target is a submarine, although any mobile naval platform could serve as the target.

The algorithmic approach to combining motion with a (possibly moving)

goal location, while avoiding local hazards, may be applicable to robotic rovers, automated landing systems, and autonomous airships. The software operates in JPL's CARACaS (Control Architecture for Robotic Agent Command and Sensing) software architecture and relies on other modules for environmental perception data and information on the predicted detectability of the target, as well as the low-level interface to the boat controls.

This work was done by Michael Wolf, Yoshiaki Kuwata, and Dimitri V. Zarzhitsky of Caltech for NASA's Jet Propulsion Laboratory. Further information is contained in a TSP (see page 1).

In accordance with Public Law 96-517, the contractor has elected to retain title to this invention. Inquiries concerning rights for its commercial use should be addressed to:

*Innovative Technology Assets Management
JPL*

Mail Stop 321-123

4800 Oak Grove Drive

Pasadena, CA 91109-8099

E-mail: iaoffice@jpl.nasa.gov

Refer to NPO-48115, volume and number of this NASA Tech Briefs issue, and the page number.

Adams-Based Rover Terramechanics and Mobility Simulator — ARTEMIS

The Mars Exploration Rovers (MERs), Spirit and Opportunity, far exceeded their original drive distance expectations and have traveled, at the time of this reporting, a combined 29 kilometers across the surface of Mars. The Rover Sequencing and Visualization Program (RSVP), the current program used to plan drives for MERs, is only a kinematic simulator of rover movement. Therefore, rover response to various terrains and soil types cannot be modeled. Although sandbox experiments attempt to model rover-terrain interaction, these experiments are time-intensive and costly, and they cannot be used within the tactical timeline of rover driving. Imaging techniques and hazard avoidance features on MER help to prevent the rover from traveling over dangerous terrains, but mobility issues have shown that these methods are not always sufficient.

ARTEMIS, a dynamic modeling tool for MER, allows planned drives to be

simulated before commands are sent to the rover. The deformable soils component of this model allows rover-terrain interactions to be simulated to determine if a particular drive path would take the rover over terrain that would induce hazardous levels of slip or sink. When used in the rover drive planning process, dynamic modeling reduces the likelihood of future mobility issues because high-risk areas could be identified before drive commands are sent to the rover, and drives planned over these areas could be rerouted.

The ARTEMIS software consists of several components. These include a preprocessor, Digital Elevation Models (DEMs), Adams rover model, wheel and soil parameter files, MSC Adams GUI (commercial), MSC Adams dynamics solver (commercial), terramechanics subroutines (FORTRAN), a contact detection engine, a soil modification engine, and output DEMs of deformed soil. The preprocessor is used to define the terrain (from a DEM) and define the soil parameters for the terrain file. The Adams rover model is placed in this terrain. Wheel and soil parameter files can be altered in the respective text files. The rover model and terrain are viewed in Adams View, the GUI for ARTEMIS. The Adams dynamics solver calls terramechanics subroutines in FORTRAN containing the Bekker-Wong equations. These subroutines use contact and soil modification engines to produce the simulation of rover movement over deformable soils, viewed in Adams View.

New drive techniques could be tested in ARTEMIS to avoid wasting limited time and energy during real-time drives. Extrication techniques can also be developed using ARTEMIS without sandbox testing. These uses of dynamic modeling are not limited to Martian vehicles, and ARTEMIS would have similar benefits for lunar vehicles. ARTEMIS could potentially be modified to dynamically simulate the movement of any vehicle over deformable soil.

This work was done by Brian P. Trease and Randel A. Lindemann of Caltech; Raymond E. Arvidson, Keith Bennett, Lauren P. Van Dyke, and Feng Zhou of the Washington University at St. Louis; and Karl Iagnemma and Carmine Senatore of MIT for NASA's Jet Propulsion Laboratory. Further information is contained in a TSP (see page 1).

This software is available for commercial licensing. Please contact Dan Broderick at Daniel.F.Broderick@jpl.nasa.gov. Refer to NPO-47781.

ISTP CDF Skeleton Editor

Basic Common Data Format (CDF) tools (e.g., cdfedit) provide no specific support for creating International Solar-Terrestrial Physics/Space Physics Data Facility (ISTP/SPDF) standard files. While it is possible for someone who is familiar with the ISTP/SPDF metadata guidelines to create compliant files using just the basic tools, the process is error-prone and unreasonable for someone without ISTP/SPDF expertise. The key problem is the lack of a tool with specific support for creating files that comply with the ISTP/SPDF guidelines. There are basic CDF tools such as cdfedit and skeletoncdf for creating CDF files, but these have no specific support for creating ISTP/SPDF compliant files.

The SPDF ISTP CDF skeleton editor is a cross-platform, Java-based GUI editor program that allows someone with only a basic understanding of the ISTP/SPDF guidelines to easily create compliant files. The editor is a simple graphical user interface (GUI) application for creating and editing ISTP/SPDF guideline-compliant skeleton CDF files. The SPDF ISTP CDF skeleton editor consists of the following components: A swing-based Java GUI program, JavaHelp-based manual/tutorial, Image/Icon files, and HTML Web page for distribution. The editor is available as a traditional Java desktop application as well as a Java Network Launching Protocol (JNLP) application. Once started, it functions like a typical Java GUI file editor application for creating/editing application-unique files.

The editor provides ease of use and support for ISTP/SPDF and project-specific standards. The editor provides support for creating/editing CDF files that comply with the ISTP/SPDF guidelines.

This work was done by Reine Chimiak and Bernard Harris of Goddard Space Flight Center, and Phillip Williams of QSS Group. Further information is contained in a TSP (see page 1). GSC-16256-1

Uplink Summary Generator (ULSGEN) Version 1.0

The Uplink Summary Generator (ULSGEN) provides a convenient means of gathering together a set of uplink related files, parsing and analyzing these

files, and producing a summary of their contents, which may then be electronically signed by one or more reviewers to verify the commands. Spacecraft operations personnel view this summary as a final sanity check before actual radiation of the uplink data.

Unique features of the software are a browser-based application that can be used both inside and outside the flight operations firewall, file retrieval from the project file server or from the project DOM (Distributed Object Manager), and the ability to parse and analyze spacecraft command files (SCMF). The software also features DSN keyword file (DKF) parsing for uplink windows, and enables hosting of one or more projects in a single server. Each project can define its own uplink summary template.

Each uplink summary is generated based on the analysis results from the parsers and the selected project template. The uplink summary review and signature collection cycle supports both parallel and sequential workflows. RadList file generation enables linkage to the command system.

This work was done by Yeou-Fang Wang, Mitchell Schrock, Timothy J. Reeve, Kristine T. Fong, and Benjamin D. Smith of Caltech for NASA's Jet Propulsion Laboratory. Further information is contained in a TSP (see page 1).

This software is available for commercial licensing. Please contact Dan Broderick at Daniel.F.Broderick@jpl.nasa.gov. Refer to NPO-48423.

Robotics On-Board Trainer (ROBoT)

ROBoT is an on-orbit version of the ground-based Dynamics Skills Trainer (DST) that astronauts use for training on a frequent basis. This software consists of two primary software groups. The first series of components is responsible for displaying the graphical scenes. The remaining components are responsible for simulating the Mobile Servicing System (MSS), the Japanese Experiment Module Remote Manipulator System (JEM-RMS), and the H-II Transfer Vehicle (HTV) Free Flyer Robotics Operations. The MSS simulation software includes: Robotic Workstation (RWS) simulation, a simulation of the Space Station Remote Manipulator System (SSRMS), a simulation of the ISS Command and Control System (CCS), and a portion of the Portable Computer System (PCS) software necessary for MSS operations.

These components all run under the CentOS4.5 Linux operating system. The JEMRMS simulation software includes real-time, HIL, dynamics, manipulator multi-body dynamics, and a moving object contact model with Tricks discrete time scheduling. The JEMRMS DST will be used as a functional proficiency and skills trainer for flight crews. The HTV Free Flyer Robotics Operations simulation software adds a functional simulation of HTV vehicle controllers, sensors, and data to the MSS simulation software. These components are intended to support HTV ISS visiting vehicle analysis and training. The scene generation software will use DOUG (Dynamic On-orbit Ubiquitous Graphics) to render the graphical scenes. DOUG runs on a laptop running the CentOS4.5 Linux operating system. DOUG is an Open GL-based 3D computer graphics rendering package. It uses pre-built three-dimensional models of on-orbit ISS and space shuttle systems elements, and provides real-time views of various station and shuttle configurations.

This work was done by Genevieve Johnson of Johnson Space Center and Greg Alexander of Harmony Lane Studios, Inc. Further information is contained in a TSP (see page 1). MSC-25005-1

Software Engineering Tools for Scientific Models

Software tools were constructed to address issues the NASA Fortran development community faces, and they were tested on real models currently in use at NASA. These proof-of-concept tools address the High-End Computing Program and the Modeling, Analysis, and Prediction Program. Two examples are the NASA Goddard Earth Observing System Model, Version 5 (GEOS-5) atmospheric model in Cell Fortran on the Cell Broadband Engine, and the Goddard Institute for Space Studies (GISS) coupled atmosphere-ocean model called ModelE, written in fixed format Fortran.

To test the tool set, the innovators first extended an annotation and conversation mechanism, known as Activities, allowing developers to provide insights into code without modifying it to include the qualification of Activities with metadata for filtering. Next, the designers created a visualization to present the relationships, or connectivity, between model variables by tracing various constructs through different components and levels of a model.