

ously were created. These KML file-based overlays could then be manually manipulated as image overlays, saved, and then uploaded to the project server where they are parsed and the metadata in the database is updated. The new interface eliminates the need to save, download, open, re-save, and upload the KML files. Everything is processed on the Web, and all manipulations go directly into the database. Administrators also have the control to discard any single correction that was made and validate a correction.

This program streamlines a process that previously required several critical steps and was probably too complex for the average user to complete suc-

cessfully. The new process is theoretically simple enough for members of the public to make use of and contribute to the success of the Sally Ride EarthKAM project.

Using the Google Earth Web plug-in, EarthKAM images, and associated metadata, this software allows users to interactively manipulate an EarthKAM image overlay, and update and improve the associated metadata. The Web interface uses the Google Earth JavaScript API along with PHP-PostgreSQL to present the user the same interface capabilities without leaving the Web. The simpler graphical user interface will allow the public to participate directly and mean-

ingfully with EarthKAM. The use of similar techniques is being investigated to place ground-based observations in a Google Mars environment, allowing the MSL (Mars Science Laboratory) Science Team a means to visualize the rover and its environment.

This work was done by Paul M. Andres of Caltech, Dennis K. Lazar of Purdue University, and Robert Q. Thames of Loyola Marymount University 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-48800.

Trade Space Specification Tool (TSST) for Rapid Mission Architecture (Version 1.2)

NASA's Jet Propulsion Laboratory, Pasadena, California

Trade Space Specification Tool (TSST) is designed to capture quickly ideas in the early spacecraft and mission architecture design and categorize them into trade space dimensions and options for later analysis. It is implemented as an Eclipse RCP Application, which can be run as a standalone program. Users rapidly create concept items with single clicks on a graphical canvas, and can organize and create linkages between the ideas using drag-and-drop actions within the same graphical view. Various views such as a trade view, rules view, and architecture view are provided to help users to visualize the trade space.

This software can identify, explore, and assess aspects of the mission trade space, as well as capture and organize

linkages/dependencies between trade space components. The tool supports a user-in-the-loop preliminary logical examination and filtering of trade space options to help identify which paths in the trade space are feasible (and preferred) and what analyses need to be done later with executable models. This tool provides multiple user views of the trade space to guide the analyst/team to facilitate interpretation and communication of the trade space components and linkages, identify gaps in combining and selecting trade space options, and guide user decision-making for which combinations of architectural options should be pursued for further evaluation.

This software provides an environment to capture mission trade space ele-

ments rapidly and assist users for their architecture analysis. This is primarily focused on mission and spacecraft architecture design, rather than general-purpose design application. In addition, it provides more flexibility to create concepts and organize the ideas. The software is developed as an Eclipse plug-in and potentially can be integrated with other Eclipse-based tools.

This work was done by Yeou-Fang Wang, Mitchell Schrock, Chester S. Borden, and Robert C. Moeller 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-48158.

Acoustic Emission Analysis Applet (AEAA) Software

John H. Glenn Research Center, Cleveland, Ohio

NASA Glenn Research and NASA White Sands Test Facility have developed software supporting an automated pressure vessel structural health monitoring (SHM) system based on acoustic emissions (AE). The software, referred to as the Acoustic Emission Analysis Applet (AEAA), provides analysts with a tool that can interrogate data collected on Digital Wave Corp. and Physical Acoustics Corp. software

using a wide spectrum of powerful filters and charts. This software can be made to work with any data once the data format is known. The applet will compute basic AE statistics, and statistics as a function of time and pressure (see figure). AEAA provides value added beyond the analysis provided by the respective vendors' analysis software. The software can handle data sets of unlimited size.

A wide variety of government and commercial applications could benefit from this technology, notably requalification and usage tests for compressed-gas and hydrogen-fueled vehicles. Future enhancements will add features similar to a "check engine" light on a vehicle. Once installed, the system will ultimately be used to alert International Space Station crewmembers to critical structural instabilities, but will have lit-



The **Front Panel** is divided into the following main sections: (1) Run-time menu; File, Options, Help; (2) AE & Fast Fourier Transform graphs; (3) AE analysis settings; and (4) AE statistics.

tle impact to missions otherwise. Diagnostic information could then be transmitted to experienced technicians on the ground in a timely manner to determine whether pressure vessels have been impacted, are structurally unsound, or can be safely used to complete the mission.

This work was done by Don J. Roth and Charles T. Nichols of Glenn Research Center, and was sponsored by the NASA Non-destructive Evaluation Working Group. Further information is contained in a TSP (see page 1).

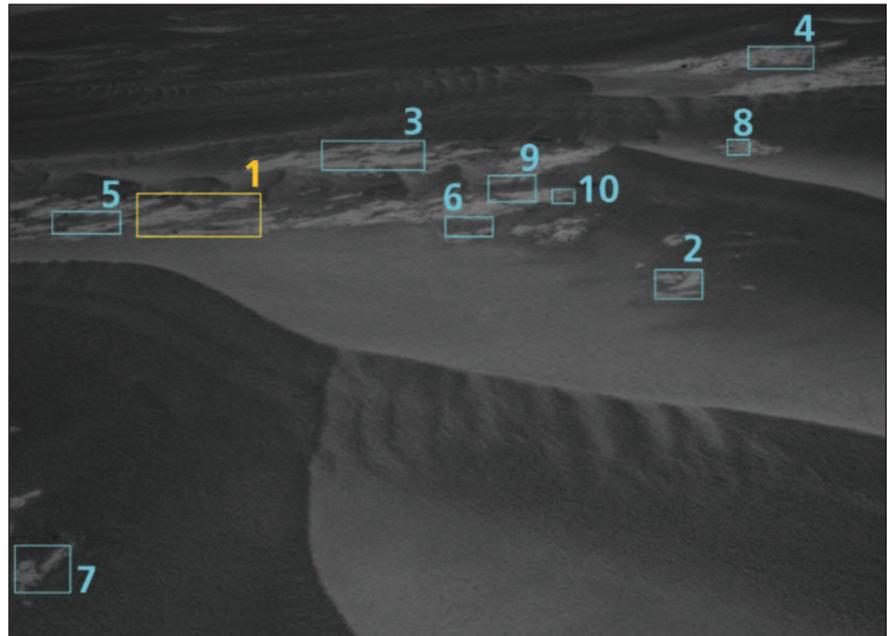
Inquiries concerning rights for the commercial use of this invention should be addressed to NASA Glenn Research Center, Innovative Partnerships Office, Attn: Steven Fedor, Mail Stop 4-8, 21000 Brookpark Road, Cleveland, Ohio 44135. Refer to LEW-19032-1

▶ Memory-Efficient Onboard Rock Segmentation

NASA's Jet Propulsion Laboratory, Pasadena, California

Rockster-MER is an autonomous perception capability that was uploaded to the Mars Exploration Rover Opportunity in December 2009. This software provides the vision front end for a larger software system known as AEGIS (Autonomous Exploration for Gathering Increased Science), which was recently named 2011 NASA Software of the Year. As the first step in AEGIS, Rockster-MER analyzes an image captured by the rover, and detects and automatically identifies the boundary contours of rocks and regions of outcrop present in the scene. This initial segmentation step reduces the data volume from millions of pixels into hundreds (or fewer) of rock contours. Subsequent stages of AEGIS then prioritize the best rocks according to scientist-defined preferences and take high-resolution, follow-up observations (see figure). Rockster-MER has performed robustly from the outset on the Mars surface under challenging conditions.

Rockster-MER is a specially adapted, embedded version of the original Rockster algorithm ("Rock Segmentation Through Edge Regrouping," (NPO-44417) *Software Tech Briefs*, September 2008, p. 25). Although the new version performs the same basic task as the original code, the software has been (1) significantly upgraded to overcome the severe onboard resource limitations (CPU,



The **top ten objects** detected by Rockster-MER running onboard the Opportunity rover on Sol 2221. Note: Rockster-MER detects and segments the objects; the ranking is provided by AEGIS according to a set of scientist-specified attribute weightings.

memory, power, time) and (2) "bullet-proofed" through code reviews and extensive testing and profiling to avoid the occurrence of faults. Because of the limited computational power of the RAD6000 flight processor on Opportunity (roughly two orders of magnitude

slower than a modern workstation), the algorithm was heavily tuned to improve its speed. Several functional elements of the original algorithm were removed as a result of an extensive cost/benefit analysis conducted on a large set of archived rover images. The algorithm was also re-