to each band in each scene. It uses a data-mask image for selecting data and blending of input scenes. Under control by a user, the program can be made to operate on small parts of the output image space, with check-point and restart capabilities. The program runs on SGI IRIX computers. It is capable of parallel processing using shared-memory code, large memories, and tens of central processing units. It can retrieve input data and store output data at locations remote from the processors on which it is executed.

*This program was written by Lucian Plesea 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 Karina Edmonds of the California Institute of Technology at (818) 393-2827. Refer to NPO-40999.*

## Architecture for Control of the K9 Rover

Software featuring a multilevel architecture is used to control the hardware on the K9 Rover, which is a mobile robot used in research on robots for scientific exploration and autonomous operation in general. The software consists of five types of modules:

- *Device Drivers* — These modules, at the lowest level of the architecture, directly control motors, cameras, data buses, and other hardware devices.
- *Resource Managers* — Each of these modules controls several device drivers. Resource managers can be commanded by either a remote operator or the pilot or conditional-executive modules described below.
- *Behaviors and Data Processors* — These modules perform computations for such functions as planning paths, avoiding obstacles, visual tracking, and stereoscopy. These modules can be commanded only by the pilot.
- *Pilot* — The pilot receives a possibly complex command from the remote operator or the conditional executive, then decomposes the command into (1) more-specific commands to the resource managers and (2) requests for information from the behaviors and data processors.
- *Conditional Executive* — This highest-level module interprets a command plan sent by the remote operator, determines whether resources required for execution of the plan are available, monitors

execution, and, if necessary, selects an alternate branch of the plan.

*This program was written by John L. Bresina and Maria Bualat of **Ames Research Center**; Michael Fair and Anne Wright of QSS Group, Inc.; and Richard Washington of Research Institute for Advanced Computer Science. Further information is contained in a TSP (see page 1).*

*Inquiries concerning rights for the commercial use of this invention should be addressed to the Ames Innovative Partnerships Office at (650) 604-2954. Refer to ARC-14587-1.*

## HFGMC Enhancement of MAC/GMC

Additional information about a mathematical model denoted the high-fidelity generalized method of cells (HFGMC) and implementation of the HFGMC within version 4.0 of the MAC/GMC software has become available. MAC/GMC (Micromechanics Analysis Code With Generalized Method of Cells) was a topic of several prior *NASA Tech Briefs* articles, version 4.0 having been described in "Comprehensive Micromechanics-Analysis Code — Version 4.0" (LEW-17495-1), *NASA Tech Briefs*, Vol. 29, No. 9 (September 2005), page 54. MAC/GMC predicts elastic and inelastic thermomechanical responses of composite materials. MAC/GMC utilizes the generalized method of cells (GMC) — a model of micromechanics that predicts macroscopic responses of a composite material as functions of the properties, sizes, shapes, and responses of its constituents (e.g., matrix and fibers). The accuracy of the GMC is limited by neglect of coupling between normal and shear stresses. The HFGMC was developed by combining elements of the GMC and a related model, denoted the higher-order theory for functionally graded materials (HOTFGM), that can account for this coupling. Hence, the HFGMC enables simulation of stress and strain with greater accuracy. Some alterations of the MAC/GMC data structure were necessitated by the greater computational complexity of the HFGMC.

*This program was written by Steven M. Arnold of **Glenn Research Center**, Jacob Aboudi and Marek-Jerzy Pindera of the University of Virginia, and Brett A. Bednarcyk of Ohio Aerospace Institute. 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: Steve Fedor, Mail Stop 4–8, 21000 Brookpark Road, Cleveland, Ohio 44135. Refer to LEW-17818-1.*

## Automated Activation and Deactivation of a System Under Test

The MPLM Automated Activation/Deactivation application (MPLM means Multi-Purpose Logistic Module) was created with a three-fold purpose in mind:

1. To reduce the possibility of human error in issuing commands to, or interpreting telemetry from, the MPLM power, computer, and environmental control systems;
2. To reduce the amount of test time required for the repetitive activation/deactivation processes; and
3. To reduce the number of on-console personnel required for activation/ deactivation.

All of these have been demonstrated with the release of the software. While some degree of automated end-item commanding had previously been performed for space-station hardware in the test environment, none approached the functionality and flexibility of this application. For MPLM activation, it provides mouse-click selection of the hardware complement to be activated, activates the desired hardware and verifies proper feedbacks, and alerts the user when telemetry indicates an error condition or manual intervention is required. For MPLM deactivation, the product senses which end items are active and deactivates them in the proper sequence. For historical purposes, an on-line log is maintained of commands issued and telemetry points monitored. The benefits of the MPLM Automated Activation/ Deactivation application were demonstrated with its first use in December 2002, when it flawlessly performed MPLM activation in 8 minutes (versus as much as 2.4 hours for previous manual activations), and performed MPLM deactivation in 3 minutes (versus 66 minutes for previous manual deactivations). The number of test team members required has dropped from eight to four, and in actuality the software can be operated by a sole (knowledgeable) system engineer.

*This work was done by Mark A. Poff of The Boeing Company for **Kennedy Space Center**. For further information, contact the Kennedy Innovative Partnerships Office at (321) 867-1463. Refer to KSC-12653.*