Mixed Real/ Virtual Operator Interface for ATHLETE
NASA's Jet Propulsion Laboratory, Pasadena, California

The mixed real/virtual operator interface for ATHLETE (MSim-ATHLETE) is a new software system for operating manipulation and inspection tasks in JPL's ATHLETE (All-Terrain Hex-Legged Extra-Terrestrial Explorer). The system presents the operator with a graphical model of the robot and a palette of available joint types. Once virtual articulations are constructed for a task, the operator can move any joint or link, and the system interactively responds in real-time with a compatible motion for all joints that best satisfies all constraints.

Unique features of the software include:
- On-line topological dynamism: The key feature of MSim-ATHLETE is that it permits the kinematic structure of the operated mechanism to be changed dynamically by the operator. These changes are not (usually) meant to indicate actual changes in the physical system, but rather add/remove virtual extensions for constraining and parameterizing motions.
- Mixed reification: MSim-ATHLETE models two kinds of articulations real articulations model the robot and virtual articulations model the virtual extensions.
- Pure kinematicity: MSim-ATHLETE is purely kinematic and thus does not require specifying any physics parameters, such as mass and friction properties.
- Useful handling of under- and over-constraint: MSim-ATHLETE allows the operator to specify both under- and over-constrained motions. In both cases, several features help organize and structure the result, including prioritized constraints, explicit hierarchical decomposition, and least-squares solving.

This work was done by Jeffrey S. Norris and David S. Millman of Caltech and Marcelle A. Vona and Daniela Rus of Massachusetts Institute of Technology for NASA's Jet Propulsion Laboratory. For more information, see http://www.mit.edu/~vona/MSim-ATHLETE/MSim-ATHLETE-info.html.

This software is available for commercial licensing. Please contact Daniel Broderick of the California Institute of Technology at danielb@caltech.edu. Refer to NPO-46869.
ing occurs during scheduled periods every day of the week, the ACR software continuously monitors the antenna equipment.

This work was done by Roger Y. Chao, Scott C. Morgan, Martha M. Strain, Stephen T. Rockwell, Kenneth J. Shimizu, Barzia J. Tehrani, Jaclyn H. Kwok, Michelle Tuason-Wong, and Henry Valtier of Caltech; Reza Nalbandi of MTC; Michael Wert of ITT; and Patrick Leung of ISDS/Averstar for NASA’s Jet Propulsion Laboratory. For more information, contact iaoffice@jpl.nasa.gov.

This software is available for commercial licensing. Please contact Daniel Broderick of the California Institute of Technology at danielb@caltech.edu. Refer to NPO-47002.

**Efficient Parallel Engineering Computing on Linux Workstations**

**NASA’s Jet Propulsion Laboratory, Pasadena, California**

A C software module has been developed that creates lightweight processes (LWPs) dynamically to achieve parallel computing performance in a variety of engineering simulation and analysis applications to support NASA and DoD project tasks. The required interface between the module and the application it supports is simple, minimal and almost completely transparent to the user applications, and it can achieve nearly ideal computing speed-up on multi-CPU engineering workstations of all operating system platforms. The module can be integrated into an existing application (C, C++, Fortran and others) either as part of a compiled module or as a dynamically linked library (DLL).

This software has the following major advantages over existing commercial and public domain software of similar functionality.

1. It is especially applicable to and powerful on commercially, widely available, multi-CPU engineering workstations;
2. It has a very simple software architecture and user interface and can be quickly integrated into an existing application; and
3. Its code size is very small, and its performance overhead is minimal, resulting in nearly ideal parallel-computing performance for many computing-intensive scientific and engineering applications.

The approach adopted in this technology development does not require any additional hardware and software beyond what’s typically available on any commercial engineering workstations, that is a native operating system and C, C++ or FORTRAN compilers that an application needs.

This work was done by John Z. Lou of Caltech for NASA’s Jet Propulsion Laboratory. For more information, contact iaoffice@jpl.nasa.gov.

The software used in this innovation is available for commercial licensing. Please contact Daniel Broderick of the California Institute of Technology at danielb@caltech.edu. Refer to NPO-46892.

**FAILSAFE Health Management for Embedded Systems**

**NASA’s Jet Propulsion Laboratory, Pasadena, California**

The FAILSAFE project is developing concepts and prototype implementations for software health management in mission-critical, real-time embedded systems. The project unites features of the industry-standard ARINC 653 Avionics Application Software Standard Interface and JPL’s Mission Data System (MDS) technology (see figure). The ARINC 653 standard establishes requirements for the services provided by partitioned, real-time operating systems. The MDS technology provides a state analysis method, canonical architecture, and software framework that facilitates the design and implementation of software-intensive complex systems. The MDS technology has been used to provide the health management function for an ARINC 653 application implementation. In particular, the focus is on showing how this combination enables reasoning about, and recovering from, application software problems.

The **FAILSAFE** model-based health management concept is depicted in the block diagram.