The model library can be used to support SysML user models in various ways. A simple approach is to define and document libraries of reusable systems of units and quantities for reuse across multiple projects, and to link units and quantity kinds from these libraries to Unit and QuantityKind stereotypes defined in SysML user models.

This work was done by Nicolas F. Rouquette of Caltech, Hans-Peter DeKoenig of the European Space Agency, Roger Burkhart of Deere & Company, and Huascar Espinoza of the French Centre of Atomic Energy 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-47251.

Sptrace

NASA's Jet Propulsion Laboratory, Pasadena, California

Sptrace is a general-purpose space utilization tracing system that is conceptually similar to the commercial "Purify" product used to detect leaks and other memory usage errors. It is designed to monitor space utilization in any sort of "heap," i.e., a region of data storage on some device (nominally memory; possibly shared and possibly persistent) with a flat address space. This software can trace usage of shared and/or non-volatile storage in addition to private RAM (random access memory).

Sptrace is implemented as a set of C function calls that are invoked from within the software that is being examined. The function calls fall into two broad classes: (1) functions that are embedded within the heap management software [e.g., JPL's SDR (Simple Data Recorder) and PSM (Personal Space Management) systems] to enable heap usage analysis by populating a virtual time-sequenced "log" of usage activity, and (2) reporting functions that are embedded within the application program whose behavior is suspect. For ease of use, these functions may be wrapped privately inside public functions offered by the heap management software. Sptrace can be used for VxWorks or RTEMS real-time systems as easily as for Linux or OS/X systems.

This work was done by Scott C. Burleigh of ACRO 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 Daniel Broderick of the California Institute of Technology at danielb@caltech.edu. Refer to NPO-41626.

S-Band POSIX Device Drivers for RTEMS

NASA's Jet Propulsion Laboratory, Pasadena, California

This is a set of POSIX device driver level abstractions in the RTEMS RTOS (Real-Time Executive for Multiprocessor Systems real-time operating system) to S-Band radio hardware devices that have been instantiated in an FPGA (field-programmable gate array). These include A/D (analog-to-digital) sample capture, D/A (digital-to-analog) sample playback, PLL (phase-locked-loop) tuning, and PWM (pulse-width-modulation)-controlled gain. This software interfaces to S-band radio hardware in an attached Xilinx Virtex-2 FPGA. It uses plug-and-play device discovery to map memory to device IDs. Instead of interacting with hardware devices directly, using direct-memory mapped access at the application level, this driver provides an application programming interface (API) offering that easily uses standard POSIX function calls. This simplifies application programming, enables portability, and offers an additional level of protection to the hardware.

There are three separate device drivers included in this package: sband_device (ADC capture and DAC playback), pll_device (RF front end PLL tuning), and pwm_device (RF front end AGC control).

This work was done by James P. Lux, Minh Lang, Kenneth J. Peters, and Gregory H. Taylor of Caltech 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-47496.

MaROS: Information Management Service

NASA's Jet Propulsion Laboratory, Pasadena, California

This software is provided by the Mars Relay Operations Service (MaROS) task to a variety of Mars projects for the purpose of coordinating communications sessions between landed spacecraft assets and orbiting spacecraft assets at Mars. The Information Management Service centralizes a set of functions previously distributed across multiple spacecraft operations teams, and as such, greatly improves visibility into the end-to-end strategic coordination process. Most of the process revolves around the scheduling of communications sessions between the spacecraft during periods of time when a landed asset on Mars is geometrically visible by an orbiting spacecraft. These "relay" sessions are used to transfer data both to and from the landed asset via the orbiting asset on behalf of Earth-based spacecraft operators.

This software component is an application process running as a Java virtual