These programs provide this reachability information in an easy-to-use format by combining the surface position and orientation, arm kinematics, instrument mounting, and instrument approach angles. This software is also integrated into the ground data system and the automated processing pipelines. It understands the EDR and RDR file formats and metadata, and products tailored for in situ surface operations.

This work was done by Robert G. Deen, Patrick C. Leger, Matthew L. Robinson, and Robert G. Bonitz 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-47731.

JPL Space Telecommunications Radio System Operating Environment

A flight-qualified implementation of a Software Defined Radio (SDR) Operating Environment for the JPL-SDR built for the CoNNeCT Project has been developed. It is compliant with the NASA Space Telecommunications Radio System (STRS) Architecture Standard, and provides the software infrastructure for STRS compliant waveform applications. This software provides a standards-compliant abstracted view of the JPL-SDR hardware platform. It uses industry standard POSIX interfaces for most functions, as well as exposing the STRS API (Application Programming Interface) required by the standard. This software includes a standardized interface for IP components instantiated within a Xilinx FPGA (Field Programmable Gate Array).

The software provides a standardized abstracted interface to platform resources such as data converters, file system, etc., which can be used by STRS standards conformant waveform applications. It provides a generic SDR operating environment with a much smaller resource footprint than similar products such as SCA (Software Communications Architecture) compliant implementations, or the DoD Joint Tactical Radio Systems (JTRS).

This work was done by James P. Lux, Minh Lang, Kenneth J. Peters, Gregory H. Taylor, Courtney B. Duncan, David S. Orozco, Ryan A. Stern, Earl R. Ahten, and Mike Girard 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 Daniel Broderick of the California Institute of Technology at danielb@caltech.edu. Refer to NPO-47766.

RFI-SIM: RFI Simulation Package

RFI-SIM simulates the RFI environment to estimate the interference from terrestrial emitters into spacecraft, or vice versa. A high-fidelity simulation of the RFI environment has been developed by employing all antenna-related and radar system-related parameters of multiple emitters, as well as that of the desired spacecraft.

In the simulation, the real-time analysis of the interference and its effects on error budgets of a desired radar system is taken into account. This provides a reliable tool for radar system design to deal with RFI issues and to evaluate the sensitivity of various parts of a radar system including antenna pattern, RF front-end and digital processing to RFI signals.

The simulator is capable of a high-fidelity, complex, and real-time simulation of RFI environment. It is flexible enough to be employed for various scenarios and for several NASA missions. RFI-SIM can perform the following in support of radar system design and performance analyses:

• Error budget analyses due to RFI on a space-borne radar system;
• Sensitivity analysis of the various radar parameters, as well as hardware specs, in the presence of RFI;
• Verification of the radar system design at several stages of RF and digital components in order to evaluate their robustness against RFI;
• Assistance in algorithm development for RFI detection and removal approach;
• Based on the available database, the RFI environment over North America at L-band has been reliably and successfully simulated and validated so it can be used for L-band space-borne radars in the RFI environment; and
• Estimation of the interference from space-borne radars into terrestrial FAA radars regarding FAA compatibility issues.

This work was done by Hirad Ghaemi and Curtis W. Chen 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 Daniel Broderick of the California Institute of Technology at danielb@caltech.edu. Refer to NPO-48565.

ION Configuration Editor

The configuration of ION (Interplanetary Overlay Network) network nodes is a manual task that is complex, time-consuming, and error-prone. This program seeks to accelerate this job and produce reliable configurations.

The ION Configuration Editor is a model-based smart editor based on Eclipse Modeling Framework technology. An ION network designer uses this Eclipse-based GUI to construct a data model of the complete target network and then generate configurations. The data model is captured in an XML file. Intrinsic editor features aid in achieving model correctness, such as field fill-in, type-checking, lists of valid values, and suitable default values. Additionally, an explicit "validation" feature executes custom rules to catch more subtle model errors. A "survey" feature provides a set of reports providing an overview of the entire network, enabling a quick assessment of the model's completeness and correctness. The "configuration" feature produces the main final result, a complete set of ION configuration files (eight distinct file types) for each ION node in the network.

This work was done by Richard L. Borgen 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-48209.

Dtest Testing Software

This software runs a suite of arbitrary software tests spanning various software languages and types of tests (unit level, system level, or file comparison tests). The dtest utility can be set to automate periodic testing of large suites of software, as well as running individual tests. It supports distributing multiple tests over multiple CPU cores, if available.

The dtest tool is a utility program (written in Python) that scans through a directory (and its subdirectories) and finds all directories that match a certain pattern (directory name starts with "test_" or "test-") and then executes any tests in that directory as described in simple configuration files. The tests are completely arbitrary and are not tied to any specific programming language. A variety of tests is available to support comparing test output files

with pre-validated versions. Dtest can be used in an automated testing environment or by an individual software developer to manually create or maintain individual tests. Dtest accumulates test results in data files that can be used for reporting test results by email or on a Web site.

At the time of creation, only unit-level testing utilities such as Junit, CppUnit, etc. existed that focused on tests for a specific language. The dtest utility generalizes these capabilities to arbitrary types of tests.

This work was done by Abhinandan Jain, Jonathan M. Cameron, and Steven Myint 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 Daniel Broderick of the California Institute of Technology at danielb@caltech.edu. Refer to NPO-48199.

⚙ IMPaCT — Integration of Missions, Programs, and Core Technologies

IMPaCT enables comprehensive information on current NASA missions, prospective future missions, and the technologies that NASA is investing in, or considering investing in, to be accessed from a common Web-based interface. It allows dependencies to be established between missions and technology, and from this, the benefits of investing in individual technologies can be determined. The software also allows various scenarios for future missions to be explored against resource constraints, and the nominal cost and schedule of each mission to be modified in an effort to fit within a prescribed budget.

The objective is to establish linkages between future missions and technologies so that a more rational technology investment program can be carried out and the benefits of technologies to missions can be explored systematically. The software manages the primary data elements of Technology Sets, Technologies, Mission Sets, Missions, Time Lines, and Funding Profiles. The software reports and graphs the interrelationships (de-pendencies) among these elements in an aggregating Portfolio.

A Portfolio in IMPaCT is a set of missions and/or mission concepts and their associated technologies that can be selected by the user for the purpose of analyzing and exploring mission scenario options. Portfolios are particularly useful for understanding how a set of missions and technologies can be accommodated in a constrained funding profile by changing launch dates and/or reducing mission costs.

IMPaCT can display this information interactively or it can also be downloaded using reporting routines to standard formats such as Adobe .pdf files, MS Excel, or MS Word. IMPaCT has been developed at JPL under NASA's Planetary Science Program Support task to aid NASA in planning and defining a viable portfolio of missions and technologies.

This work was done by Carlos P. Balacuit, James A. Cutts, Craig E. Peterson, Patricia M. Beauchamp, Susan K. Jones, Winnie N. Hang, and Shahin D. Dastur of Caltech for NASA's Jet Propulsion Laboratory. For more information, go to the IMPaCT web site: https://impacts.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-48197.

⚙ Integrated Systems Health Management (ISHM) Toolkit

A framework of software components has been implemented to facilitate the development of ISHM systems according to a methodology based on Reliability Centered Maintenance (RCM). This framework is collectively referred to as the Toolkit and was developed using General Atomics' HealthMAP™ technology.

The toolkit is intended to provide assistance to software developers of mission-critical system health monitoring applications in the specification, implementation, configuration, and deployment of such applications. In addition to software tools designed to facilitate these objectives, the toolkit also pro-vides direction to software developers in accordance with an ISHM specification and development methodology. The development tools are based on an RCM approach for the development of ISHM systems. This approach focuses on defining, detecting, and predicting the likelihood of system functional failures and their undesirable consequences.

The toolkit provides users with an object-oriented environment in which to specify and program software application behavior that leverages model-based reasoning specifically targeted for ISHM applications. Furthermore, the application has been designed to follow a recommended RCM-based ISHM system design methodology, providing guidance to the developer in building the overall capability of the ISHM system. The advantages of the ISHMToolkit include: (1) guidance to ISHM system developers based on a proven methodology that strives to detect, diagnose, and predict those system failures that interfere with mission objectives; (2) access to reusable class libraries and behaviors; (3) the ability to leverage model-based reasoning; (4) the incorporation of graphical programming capabilities; (5) access to a central supervisory software layer that operates and correlates over aggregated information; and (6) a layered ISHM architecture that conforms to Open System Architecture standards.

The toolkit is a software environment designed for leveraging reusable libraries developed by General Atomics that provide generic class definition, generic class behavior, and generic failure models. The toolkit also provides capability for building or extending such class libraries.

This work was done by Meera Venkatesh, Ravi Kapadia, Mark Walker, and Kim Wilkins of General Atomics for Stennis Space Center. Inquiries concerning rights for its commercial use should be addressed to:

General Atomics
3550 General Atomics Court
San Diego, CA 92121
Telephone No. (858) 676-7169
Refer to SSC-00367, volume and number of this NASA Tech Briefs *issue, and the page number.*