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.