mapped landmarks generated per image allow for automatic detection and elimination of bad matches. Attitude and position can be generated from each image; this image-based attitude measurement can be used by the onboard navigation filter to improve the attitude estimate, which will improve the position estimates.

The algorithm uses normalized correlation of grayscale images, producing precise, sub-pixel images. The algorithm has been broken into two sub-algorithms: (1) FFT Map Matching (see figure), which matches a single large template by correlation in the frequency domain, and (2) Mapped Landmark Refinement, which matches many small templates by correlation in the spatial domain. Each relies on feature selection, the homography transform, and 3D image correlation. The algorithm is implemented in C++ and is rated at Technology Readiness Level (TRL) 4.

This work was done by Andrew Johnson, Adnan Ansar, and Larry Matthies of Caltech for NASA's Jet Propulsion Laboratory. Further information is contained in a TSP (see page 1).

The software used in this innovation is available for commercial licensing. Please contact Karina Edmonds of the California Institute of Technology at (626) 395-2322. Refer to NPO-44463.

WMAP C&DH Software

Goddard Space Flight Center, Greenbelt, Maryland

The command-and-data-handling (C&DH) software of the Wilkinson Microwave Anisotropy Probe (WMAP) spacecraft functions as the sole interface between (1) the spacecraft and its instrument subsystem and (2) ground operations equipment. This software includes a command-decoding and -distribution system, a telemetry/data-handling system, and a data-storage-and-playback system. This software performs onboard processing of attitude sensor data and generates commands for attitude-control actuators in a closed-loop fashion. It also processes stored commands and monitors health and safety functions for the spacecraft and its instrument subsystems. The basic functionality of this software is the same of that of the older C&DH software of the Rossi X-Ray Timing Explorer (RXTE) spacecraft, the main difference being the addition of the attitude-control functionality. Previously, the C&DH and attitude-control computations were performed by different processors because a single RXTE processor did not have enough processing power. The WMAP spacecraft includes a more-powerful processor capable of performing both computations.

This program was written by Alan Cudmore, Tim Leath, Art Ferrer, Todd Miller, Mark Walters, Bruce Savadkin, and Ji-Wei Wu of Goddard Space Flight Center; Steve Slegel of Daedalian Systems Corp.; and Emory Stagmer of Litton/PRC. Further information is contained in a TSP (see page 1).
GSC-14964-1

Web-Based Environment for Maintaining Legacy Software

Lyndon B. Johnson Space Center, Houston, Texas

"Advanced Tool Integration Environment" ("ATIE") is the name of both a software system and a Web-based environment created by the system for maintaining an archive of legacy software and expertise involved in developing the legacy software. ATIE can also be used in modifying legacy software and developing new software. The information that can be encapsulated in ATIE includes experts' documentation, input and output data of tests cases, source code, and compilation scripts. All of this information is available within a common environment and retained in a database for ease of access and recovery by use of powerful search engines. ATIE also accommodates the embedment of supporting software that users require for their work, and even enables access to supporting commercial-off-the-shelf (COTS) software within the flow of the experts' work.

The flow of work can be captured by saving the sequence of computer programs that the expert uses. A user gains access to ATIE via a Web browser. A modern Web-based graphical user interface promotes efficiency in the retrieval, execution, and modification of legacy code. Thus, ATIE saves time and money in the support of new and pre-existing programs.

This program was written by Michael Tigges of Johnson Space Center; Nelson Thompson, Mark Orr, and Richard Fox of Dynacs, Inc.; and Rick Rohan of Lockheed Martin Corp. Further information is contained in a TSP (see page 1). MSC-23810-1

Information Metacatalog for a Grid

Ames Research Center, Moffett Field, California

SWIM is a Software Information Metacatalog that gathers detailed information about the software components and packages installed on a grid resource. Information is currently gathered for Executable and Linking Format (ELF) executables and shared libraries, Java classes, shell scripts, and Perl and Python modules. SWIM is built on top of the POUR framework, which is described in the preceding article. SWIM consists of a set of Perl modules for extracting software information from a system, an XML schema defining the format of data that can be added by users, and a POUR XML configuration file that describes how these elements are used to generate pe-

riodic, on-demand, and user-specified information.

Periodic software information is derived mainly from the package managers used on each system. SWIM collects information from native package managers in FreeBSD, Solaris, and IRX as well as the RPM, Perl, and Python package managers on multiple platforms. Because not all software is available, or installed in package form, SWIM also crawls the set of relevant paths from the File System Hierarchy Standard that defines the standard file system structure used by all major UNIX distributions. Using these two techniques, the vast majority of software installed on a system can be located. SWIM computes the same information gathered by the periodic routines for specific files on specific hosts, and locates software on a system given only its name and type.

This program was written by Paul Kolano of Advanced Management Technology for Ames Research Center. For further information, access http://opensource.arc.nasa.gov/ or contact the Ames Technology Partnerships Division at (650) 604-2954.
ARC-15469-1

Grid Task Execution

Ames Research Center, Moffett Field, California

IPG Execution Service is a framework that reliably executes complex jobs on a computational grid, and is part of the IPG service architecture designed to support location-independent computing. The new grid service enables users to describe the platform on which they need a job to run, which allows the service to locate the desired platform, configure it for the required application, and execute the job. After a job is submitted, users can monitor it through periodic notifications, or through queries.

Each job consists of a set of tasks that performs actions such as executing applications and managing data. Each task is executed based on a starting condition that is an expression of the states of other tasks. This formulation allows tasks to be executed in parallel, and also allows a user to specify tasks to execute when other tasks succeed, fail, or are canceled.

The two core components of the Execution Service are the Task Database, which stores tasks that have been submitted for execution, and the Task Manager, which executes tasks in the proper order, based on the user-specified starting conditions, and avoids overloading local and remote resources while executing tasks.

This program was written by Chaumin Hu of Advanced Management Technology for Ames Research Center. For further infomation,access http://opensource.arc.nasa.gov/ or contact the Ames Technology Partnerships Division at (650) 604-2954.
ARC-15529-1

Parallel-Processing Software for Correlating Stereo Images

NASA's Jet Propulsion Laboratory, Pasadena, California

A computer program implements parallel-processing algorithms for correlating images of terrain acquired by stereoscopic pairs of digital stereo cameras on an exploratory robotic vehicle (e.g., a Mars rover). Such correlations are used to create three-dimensional computational models of the terrain for navigation. In this program, the scene viewed by the cameras is segmented into subimages. Each subimage is assigned to one of a number of central processing units (CPUs) operating simultaneously. Because each subimage is smaller than a full image, the correlation process takes less time than it would if full images were processed on one CPU. Segmentation and parallelization also make the process more robust in that the smaller subimages present fewer opportunities for a correlation algorithm to "get lost" and thereby fail to converge on a solution. The effectiveness of this program has been demonstrated on several parallel-processing computer systems. Whereas correlation processing of a typical stereoscopic pair of test images on a single CPU was found to take on the order of one hour, parallel processing of the same images on a 16-CPU cluster was found to take about 3 minutes.

This program was written by Gerhard Klimeck, Robert Deen, Michael Mcauley, and Eric De Jong of Caltech for NASA's Jet Propulsion Laboratory.

This software is available for commercial licensing. Please contact Karina Edmonds of the California Institute of Technology at (626) 395-2322. Refer to NPO-30631.

Knowledge Base Editor (SharpKBE)

NASA's Jet Propulsion Laboratory, Pasadena, California

The SharpKBE software provides a graphical user interface environment for domain experts to build and manage knowledge base systems. Knowledge bases can be exported/translated to various target languages automatically, including customizable target languages. The tool enhances current practices by minimizing reliance on toolsmiths for system workflow management, and also improves the quality and maintenance of those systems by reducing the number of errors within the knowledge bases. This tool's primary capability is in the area of expert systems modeling, specifically where there is a need to capture and efficiently manage large quantities of domain information (see figure).

The SharpKBE supports C# and SHINE targets, and in concert with SHINE additionally produces C and