Ensemble: an Architecture for Mission-Operations Software

Several issues are addressed by capitalizing on the Eclipse open-source software framework.

NASA’s Jet Propulsion Laboratory, Pasadena, California

“Ensemble” is the name of an open architecture for, and a methodology for the development of, spacecraft-operations software. Ensemble is also potentially applicable to the development of non-spacecraft mission-operations-type software.

Ensemble capitalizes on the strengths of the open-source Eclipse software and its architecture to address several issues that have arisen repeatedly in the development of mission-operations software: Heretofore, mission-operations application programs have been developed in disparate programming environments and integrated during the final stages of development of missions. The programs have been poorly integrated, and it has been costly to develop, test, and deploy them. Users of each program have been forced to interact with several different graphical user interfaces (GUIs). Also, the strategy typically used in integrating the programs has yielded serial chains of operational software tools of such a nature that during use of a given tool, it has not been possible to gain access to the capabilities afforded by other tools. In contrast, the Ensemble approach offers a low-risk path towards tighter integration of mission-operations software tools.

Ensemble is based on an adaptation of the Eclipse Rich Client Platform (RCP), which is a widely used, readily available, stable, supported software framework for component-based development of application programs. The Eclipse RCP is a set of Java classes that define an architecture for general component-based application programs. New application programs are built on top of the RCP as a set of components, called plug-ins, that augment and extend its functionality. For example, a mission-activity-planning application program would consist of the RCP plus a set of plug-ins responsible for displaying, editing, and modeling activity plans. Application programs built on top of the RCP also gain access to a variety of such generally applicable capabilities as a help system, an update manager, and an extensible GUI.

In Ensemble, the difficulties of establishing interfaces between different software tools are minimized by developing most of the tools as Eclipse plug-ins. In addition, Ensemble draws upon capabilities provided by the Eclipse RCP to document and enforce interfaces between different components. In some cases, it may not be possible or prudent to develop a tool as an Eclipse Java plug-in. Such a tool can still be integrated with the Ensemble architecture. Development of a general, robust method of integrating non-Eclipse tools with other Ensemble tools is proceeding.

In Ensemble, the Eclipse framework provides a common GUI that can accommodate GUI components from multiple software tools developed by different teams. To the user, the resulting GUI looks as though it belongs to a single such tool while drawing on the resources of many of them. Ensemble provides for a task-oriented GUI that is based heavily upon an Eclipse perspective, which defines which GUI components are visible to a user at a particular time. As a user moves through tasks required for planning mission operations, the user clicks through a set of icons devoted to each task.

The chosen prior feature-based method is known as adaptive principal-component analysis (APCA); the chosen prior color-based method is known as adaptive color segmentation (ACOSE). These methods are made to interact with each other in a closed-loop system.
In principle, the improved method could readily be implemented in integrated circuitry to make a compact, low-power, real-time object-recognition system. It has been proposed to demonstrate the feasibility of such a system by integrating a 256-by-256 active-pixel sensor with APCA, ACOSE, and neural processing circuitry on a single chip. It has been estimated that such a system on a chip would have a volume no larger than a few cubic centimeters, could operate at a rate as high as 1,000 frames per second, and would consume in the order of milliwatts of power.

This work was done by Tuan Duong, Vu Duong, and Allen Stubberud of Caltech for NASA's Jet Propulsion Laboratory. Further information is contained in a TSP (see page 1).

In accordance with Public Law 96-517, the contractor has elected to retain title to this invention. Inquiries concerning rights for its commercial use should be addressed to:

Innovative Technology Assets Management
JPL
Mail Stop 202-233
4800 Oak Grove Drive
Pasadena, CA 91109-8099
(818) 354-2240
E-mail: iaoffice@jpl.nasa.gov

Refer to NPO-41370, volume and number of this NASA Tech Briefs issue, and the page number.