

base IPL with PL/SQL packages. Since it was developed using object-oriented programming in a modular fashion, it has proved easy to maintain and its capabilities are as easily extended. IMPACT has shown a very high reliability factor as well.

IMPACT manages a rapidly changing flight sequence, manifests, and detailed parts list for ISS by featuring views of an ISS IPL based on flight phases (i.e., launch, on-orbit, and return) and flights. It also features resource data viewing for each part in the IPL and a hypertext-based help system. IMPACT can be started in "view only" as well as in "update modes." When in update mode, IMPACT supports the creation of data-

base entries for new flights, elements, subelements, and parts as well as parts movement around assembly hierarchies, using menu-driven commands and buttons, and drag-and-drop technology. More than one IMPACT session can be brought up independent of another, and different views can be placed side-by-side on the same screen during the same session.

IMPACT is therefore a unique, flexible tool with an easy-to-use, highly intuitive graphical user interface. Its novelty lies in the fact that it allows users to view and manipulate IPL hierarchical data efficiently, something no other tool has allowed during the time of this reporting. Already in use on the ISS, IMPACT has

proven to be flexible and can mature and grow with a system. As such, it is a valuable adjunct not only to the space industry for which it was developed but, with suitable modifications, to large commercial databases.

This work was done by Bobby Jain, Bill Morris, and Kelly Sharpe of Barrios Technology for Johnson Space Center. For further information, contact

*Barrios Technology, Inc.
2525 Bay Area Blvd., Suite 300
Houston, TX 77058-1556
Phone: (281) 280-1900
Fax: (281) 280-1901*

Refer to MSC 22915, volume and number of this NASA Tech Briefs issue, and the page number.

➤ An Architecture for Controlling Multiple Robots

Hierarchies of behaviors can be constructed and coordinated with great versatility.

NASA's Jet Propulsion Laboratory, Pasadena, California

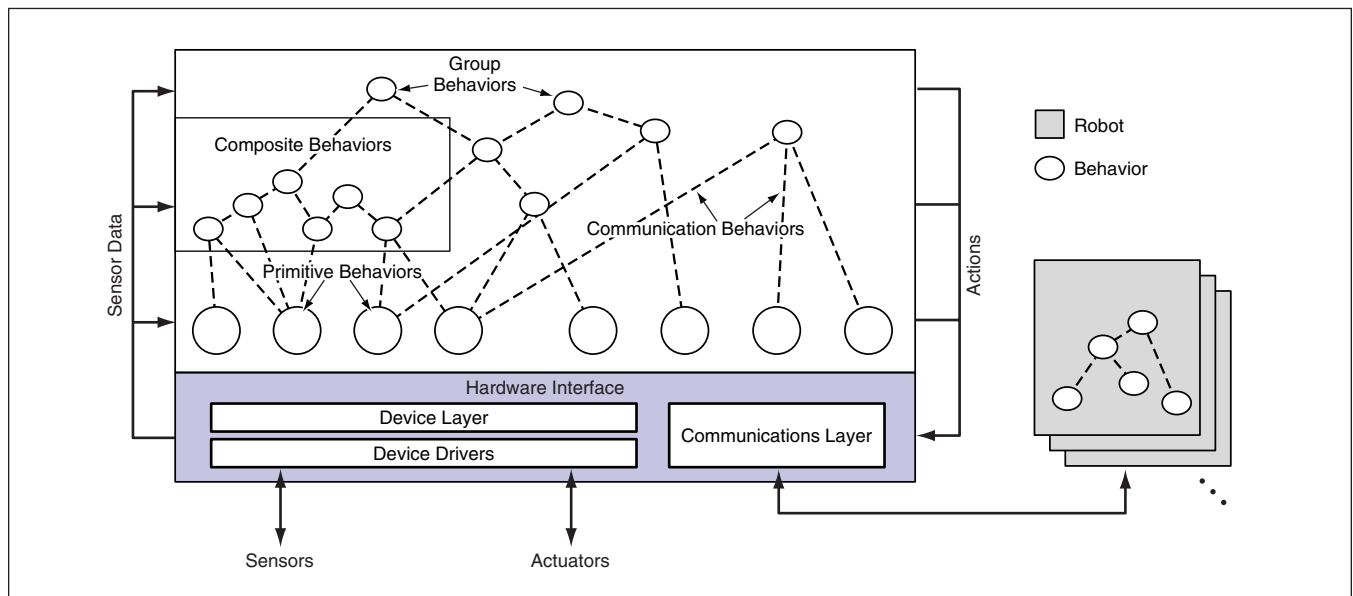
The Control Architecture for Multirobot Outpost (CAMPOUT) is a distributed-control architecture for coordinating the activities of multiple robots. In the CAMPOUT, multiple-agent activities and sensor-based controls are derived as group compositions and involve coordination of more basic controllers denoted, for present purposes, as behaviors.

The CAMPOUT provides basic mechanistic concepts for representation and execution of distributed group activities. One considers a network of nodes that com-

prise behaviors (self-contained controllers) augmented with hyper-links, which are used to exchange information between the nodes to achieve coordinated activities. Group behavior is guided by a scripted plan, which encodes a conditional sequence of single-agent activities. Thus, higher-level functionality is composed by coordination of more basic behaviors under the downward task decomposition of a multi-agent planner (see figure).

Robotics is a highly multidisciplinary field that requires efficient integration of

many components (e.g., perception, mapping, localization, control, and learning) that involve diverse representations, frameworks, and paradigms (e.g., classical control theory, artificially intelligent planners, estimation theory, data fusion, computer vision, utility theory, decision theory, fuzzy logic, and multiple-objective decision making). The CAMPOUT provides a conceptual infrastructure for consolidating diverse techniques to enable the efficient use and integration of these components for meaningful interaction and operation.



The CAMPOUT Provides for a Hierarchical Organization of primitive behaviors, composite behaviors built from primitive behaviors, and groups composed from coordination of behaviors across multiple robots. Each robot runs an instance of this architecture and participates in coordination of activities through group behaviors. Coordination is facilitated through communication behaviors.

The methodology of the CAMPOUT features a few elementary architectural mechanisms for (a) behavior representation, (b) behavior composition, (c) group coordination of teams, and (d) interfaces among (a), (b), and (c). For the purposes of the CAMPOUT, a behavior is defined and represented as a mapping from a percept (defined here as a description of raw or processed sensory input) or a sequence of percepts to an action or sequence of actions. The mapping assigns, to each possible action, a degree of preference that ranges from 0 for most undesired to 1 for most desired. This definition of a behavior is a general recipe that does not dictate how the mapping is to be implemented. It does not exclude implementation by use of a look-up table, a finite-state machine, a neural network, an expert system, a control law, or any other such means. Each behavior can be implemented using whichever approach is appropriate.

Behavior composition is the mechanism used for building higher-level behaviors by combining lower-level ones. The activities of lower-level behaviors are coordinated within the context of the task and

objective of a higher-level behavior. An explicit design goal of the CAMPOUT has been to support not one but an arbitrary number of behavior-coordination mechanisms (BCMs). The architecture can be extended by incorporation of new BCMs. Because different BCMs often require different behavior representations, the CAMPOUT involves utilization of a multi-valued behavior representation that is general enough for a large class of applications. BCMs can be divided into two main classes: arbitration and command. The CAMPOUT supports both classes.

In the CAMPOUT, the problem of coordinating a group of robots is formulated as one of coordinating multiple distributed behaviors across a network that includes more than one decision maker. In behavior coordination, one is basically concerned with resolving or managing conflicts between mutually exclusive alternatives and between behavioral objectives. Because this is as true for individual as for group decision-making, the difference between individual and group decision-making is inessential, and both can be studied in the same

conceptual framework. Mechanisms that are typically used for coordination of the behavior of one robot can then be used for coordination of behaviors running on a network of robots. Hence, for example, a control loop could use sensors on one robot to drive a different robot.

This work was done by Hrand Aghazarian, Paolo Pirjanian, Paul Schenker, and Terrance Huntsberger 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-30345, volume and number of this NASA Tech Briefs issue, and the page number.