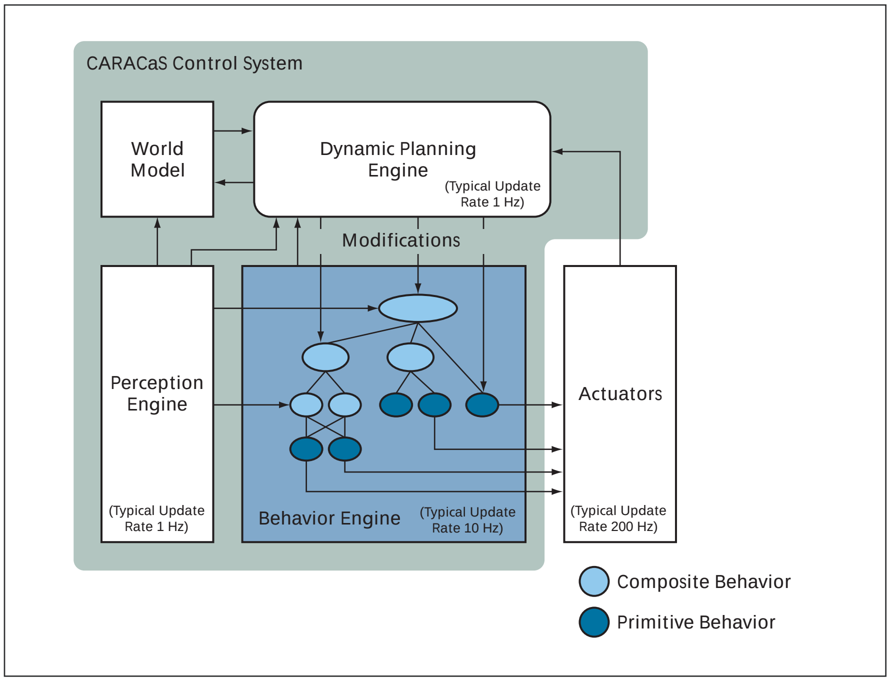
Technology Focus: Sensors

Control Architecture for Robotic Agent Command and Sensing

Plans and behaviors are updated in response to changing requirements and conditions.

NASA's Jet Propulsion Laboratory, Pasadena, California



A **CARACaS Control System** includes three coupled agents (the engines) and a world model. The network in the behavior engine is built from primitive and composite behaviors. The dynamic planning engine interacts with the network at both the primitive and composite levels.

Control Architecture for Robotic Agent Command and Sensing (CARACaS) is a recent product of a continuing effort to develop architectures for controlling either a single autonomous robotic vehicle or multiple cooperating but otherwise autonomous robotic vehicles. CARACaS is potentially applicable to diverse robotic systems that could include aircraft, spacecraft, ground vehicles, surface water vessels, and/or underwater vessels.

CARACaS (see figure) incudes an integral combination of three coupled agents: a dynamic planning engine, a behavior engine, and a perception engine. The perception and dynamic planning engines are also coupled with a memory in the form of a world model. CARACaS is intended to satisfy the need for two major capabilities essential for proper functioning of an autonomous robotic system: a capability for deterministic reaction to unanticipated occurrences and a capability for re-planning in the face of changing goals, conditions, or resources.

The behavior engine incorporates the multi-agent control architecture, called "CAMPOUT," described in "An Architecture for Controlling Multiple Robots" (NPO-30345), *NASA Tech Briefs*, Vol. 28, No. 11 (November 2004), page 65. CAMPOUT is used to develop behavior-composition and -coordination mechanisms. Real-time process algebra operators are used to compose a behavior network for any given mission scenario. These operators afford a capability for producing a formally correct kernel of behaviors that guarantee predictable performance. By use of a method based on multi-objective decision theory (MODT), recommendations from multiple behaviors are combined to form a set of control actions that represents their consensus. In this approach, all behaviors contribute simultaneously to the control of the robotic system in a cooperative rather than a competitive manner. This approach guarantees a solution that is "good enough" with respect to resolution of complex, possibly conflicting goals within the constraints of the mission to be accomplished by the vehicle(s). CARACaS further uses another MODT-based method to systematically narrow the set of possible solutions, thereby producing an output within a time orders of magnitude shorter than would be necessary to compute a solution through brute-force search of the action space.

The dynamic planning engine incorporates the architecture embodied in the CASPER software, which was described in "Software for Continuous Replanning During Execution" (NPO-20972), *NASA Tech Briefs*, Vol. 26, No. 4 (April 2002), page 67. Given an input set of mission goals and the autonomous vehicle's current state, CASPER generates a plan of activities that satisfies as many goals as possible while still obeying relevant resource constraints and operation rules. Plans are dynamically updated by use of an iterative repair algorithm that classifies conflicts (e.g., over-subscription of resources) and resolves them individually by performing one or more plan modifications. CARACaS takes a most-committed, local, heuristic, iterative repair approach to producing and modifying plans. This approach gives CARACaS the advantages of (1) enabling the application of the repair algorithm, at any time, to any given plan at any level of abstraction or detail; (2) enabling rapid replanning when conditions or goals change; (3) facilitating incorporation of heuristics to prune the search space; and (4) incurring relatively low computational overhead during search inasmuch as a local repair algorithm does not require saving of intermediate plans or backtracking points.

This work was done by Terrance Huntsberger, Hrand Aghazarian, Tara Estlin, and Daniel Gaines 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-43635.