American Nuclear Society, 9th Emergency Preparedness and Response
11th Robotics and Remote Systems for Hazardous Environments, February 11-16, 2006, Salt Lake City

Intelligent Control and Tactical Behavior Development: A long term NIST Partnership with the Army

James Albus and Anthony Barbera
Intelligent Systems Division
National Institute of Standards and Technology
Gaithersburg, Maryland 20899
James.Albus@nist.gov
Anthony.Barbera@nist.gov

Abstract – For more than 15 years, the Intelligent Systems Division of the National Institute of Standards and Technology (NIST) has worked closely with the Army Research Laboratory (ARL) on architectures and technology development for intelligent control of unmanned vehicle systems. During the early 1990's, NIST developed techniques for retrotraverse of trajectories recorded using teleoperation.

In the late 1990's NIST worked with ARL and the German Ministry of Defense to develop the 4-dimensional Real-time Control System (4D/RCS) reference model architecture for the ARL Demo III program. 4D/RCS consists of a multi-layered multi-resolutional hierarchy of computational nodes (i.e., organizational units) each containing elements of sensory processing (SP), world modeling (WM), value judgment (VJ), and behavior generation (BG). At the lower echelons, these nodes generate goal-seeking reactive behavior. At higher echelons, they enable goal-defining deliberative behavior. Throughout the hierarchy, interaction between SP, WM, VJ, and BG give rise to perception, cognition, imagination, reasoning, planning, and control.

In 2002 and 2003, NIST worked with ARL to conduct a series of engineering tests to demonstrate that autonomous mobility technology developed under the Demo III program satisfied the requirements for Technology Readiness Level 6 (i.e., a prototype demonstration in a relevant environment.)

More recently, NIST has worked with ARL to develop an engineering methodology and software development environment for endowing autonomous ground and air vehicles with the capability for tactical behaviors.

I. INTRODUCTION

A block diagram of a 4D/RCS reference model architecture [1, 2, 3] is shown in Figure 1. The architecture consists of a multi-layered multi-resolutional hierarchy of computational nodes, each containing elements of sensory processing (SP), world modeling (WM), value judgment (VJ), behavior generation (BG), and a knowledge database (KD) (included in the WM in Figure 1.) BG plans and executes actions. SP transforms sensor data into perceived and classified objects, events, and situations. WM maintains the Knowledge Database that is the node's current best estimate of the world at the scale and resolution that is appropriate for BG planner and executor processes. The WM also generates predictions – both for BG planning and SP recursive estimation. VJ evaluates the costs and benefits of predicted results of simulated plans for BG. VJ also computes the level of confidence assigned to information extracted from sensory input by SP, and assigns worth to perceived objects, events, and situations stored in KD. Each node in the architecture represents an operational unit in an organizational hierarchy.

4D/RCS computational nodes are organized such that the BG processes form a chain of command. There are horizontal communication pathways within nodes, and information in the knowledge database is shared between WM processes in nodes above, below, and at the same level within the same BG subtree. On the right, are examples of the functional characteristics of the BG processes at each level. On the left, are examples of the scale of maps generated by the SP processes and populated by the WM in the KD at each level. Sensory data paths flowing up the SP hierarchy typically form a graph, not a tree. VJ processes are hidden behind WM processes in the diagram. A control loop is typically closed at every node. An operator interface may provide input to, and obtain output from, processes in every node. (Numerical values are representative examples only. Actual numbers depend on parameters of specific system dynamics.)

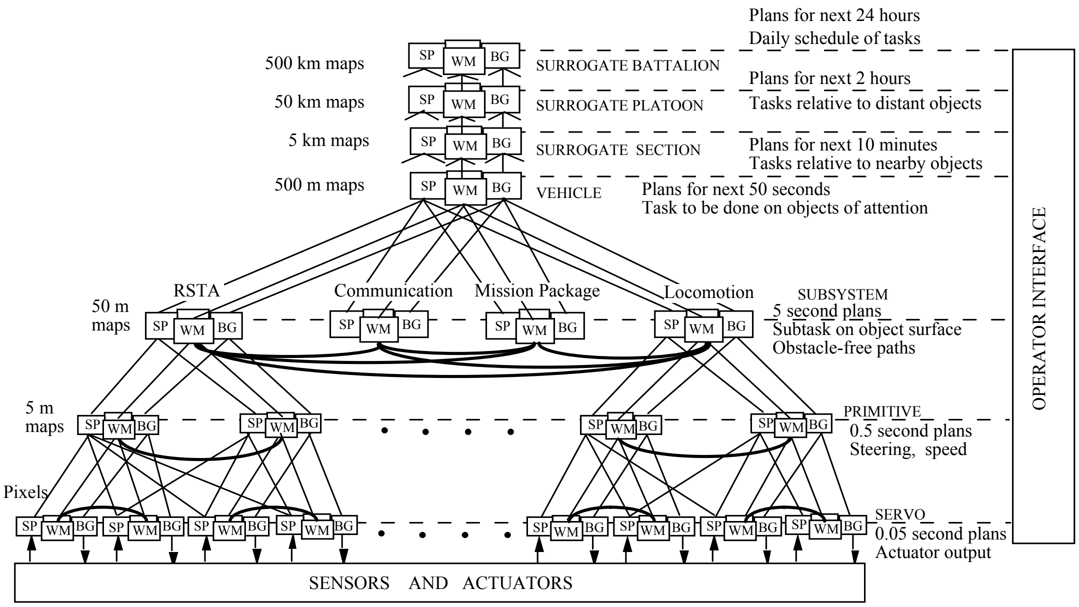

Figure 1. A 4D/RCS reference model architecture for autonomous ground vehicles

When 4D/RCS is applied to a system-of-systems, there is a node for each organizational unit in the chain of command. In the Army Future Combat System for example, at the Vehicle echelon, there is a 4D/RCS node for each vehicle. At the Section echelon, there is a node for each section, at the Platoon echelon a node for each platoon, and at the Battalion echelon a node for each battalion. Duties and responsibilities at these upper echelons involve tactics, techniques, and procedures that require a significant degree of interaction between manned and unmanned vehicles, and a great deal of sophistication in perception, situation understanding, and tactical reasoning. Plans are developed for tasks lasting minutes to hours. 4D/RCS nodes at the Section, Platoon, and Battalion echelons are currently implemented by human commanders and their staff, using computer-assisted data fusion, situation assessment, modeling, simulation, and planning systems. Tactics are developed that make best use of unmanned ground and air vehicles by assisting manned vehicles to maximize effectiveness and minimize causalities.

Within each vehicle, there are surrogate nodes for the Section, Platoon, and Battalion echelons. These provide each vehicle with higher echelon plans, models, goals, and priorities during those periods of time when the vehicle is not in direct communications with its supervisor. For Army vehicles on the ground, this frequently occurs. Surrogate nodes enable single vehicles or groups of autonomous vehicles to cooperate effectively and act appropriately, even when contact with human operators is interrupted.

2. THE INTELLIGENT NODE

Figure 2 shows a first level of detail in a typical 4D/RCS node. In each node, a behavior generation process accepts task commands with goals and parameters from a behavior generation process at the next higher echelon, and issues commanded actions with subgoals and parameters to one or more behavior generation processes at the next lower echelon. (Solid lines indicate normal data pathways. Dotted lines indicate channels by which an operator can peek and poke at data, and insert or modify control commands.)

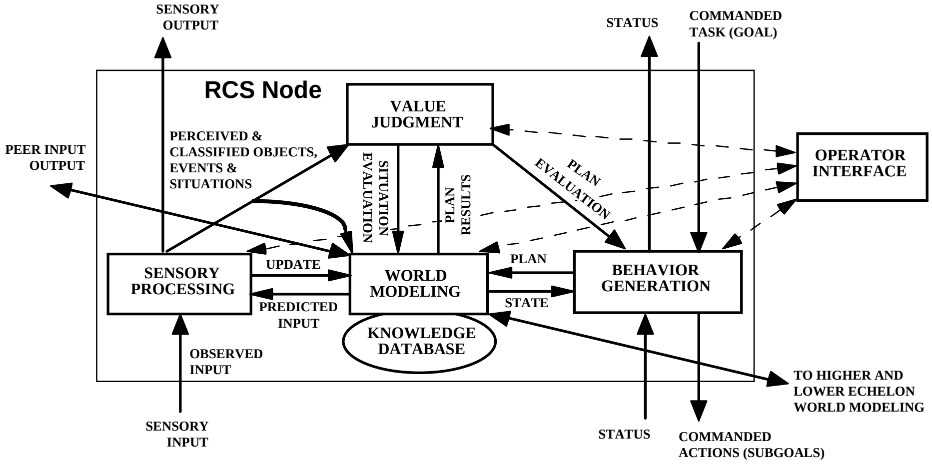

Figure 2. Internal structure of a 4D/RCS node.

Each node reports on the status of its current task to its supervisor, and receives status reports from each of its subordinates. Within the node, tentative plans are submitted by BG to WM where expected results are simulated. These are sent to VJ for evaluations that are returned to BG for decision making. This is a planning loop.

Each node also receives sensory input from lower levels of the SP hierarchy. This is compared with predicted input from WM. Variance between predictions and observations are used by WM to update KD. This is a recursive estimation loop. SP algorithms in each node also perform attention, segmentation, and classification operations. Perceived and classified entities, events, and situations are evaluated by VJ and entered into the KD by WM. Output from SP is forwarded up the SP hierarchy.

Figure 3 shows a second level of detail in a typical 4D/RCS node. The BG process accepts tasks and plans and executes behavior designed to accomplish those tasks. The internal structure of the BG process consists of a planner and a set of executors (EX). At the upper right of Figure 3, task commands from a supervisor BG process are input. A planner module decomposes each task into a set of coordinated plans for subordinate BG processes. A VJ process evaluates expected results of tentative plans. For each subordinate there is an Executor that issues commands, monitors progress, and compensates for errors between desired plans and observed results. The Executors use feedback to react quickly to emergency conditions with reflexive actions. Predictive capabilities provided by the WM may enable the Executors to eliminate delays in the feedback loop, and even generate pre-emptive behavior.

SP processes operate on input from lower echelons and sensors by windowing (i.e., focusing attention), grouping (i.e., segmentation), computing and filtering group attributes (i.e., recursive estimation), and classifying (e.g., recognizing) entities, events, patterns, and situations. VJ processes assign confidence and worth to entities, events, and situations entered into the KD.

Figure 3 illustrates that each node contains both a deliberative and a reactive component. Top-down, each node generates and executes plans designed to satisfy task goals, priorities, and constraints conveyed by commands from above. Bottom-up, each node closes a reactive control loop driven by feedback from sensors. Thus, within each node, deliberative plans are merged with reactive behaviors. Whenever the planner develops a new plan, it is substituted for the old plan in the plan buffers of the Executors. Each Executor treats its current plan as a reference trajectory. It uses planned actions as feedforward control signals, and uses the difference between planned states and estimated states as feedback control signals.

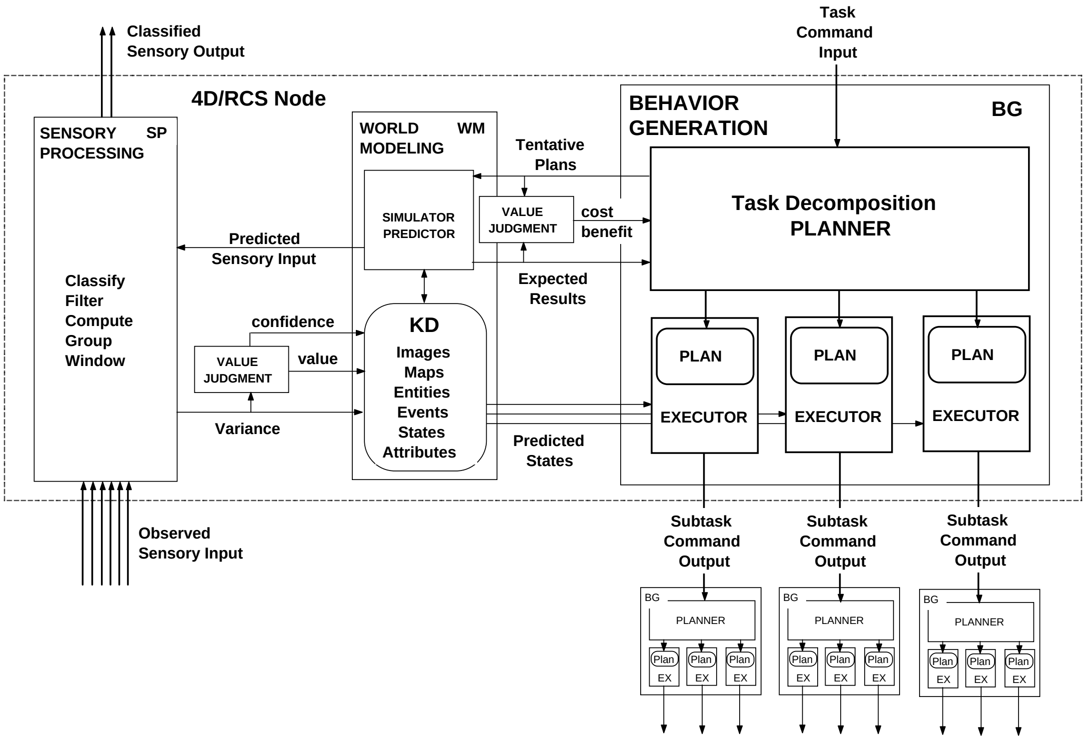

Figure 3. A typical 4D/RCS computational node.

III. 4D/RCS FOR DEMO III

During the Demo III program, [4] only the first three echelons in the locomotion subsystem of the 4D/RCS reference model architecture were implemented. Most of the emphasis was focused on the Subsystem echelon where terrain maps were overlaid with traversability costs, and planning algorithms were used to search for the lowest cost path out to the visual horizon provided by the LADAR camera. Higher level path planning using Digital Terrain Databases was performed in the Human-Robot Interface (HRI). Plans generated in the HRI were sent to the Demo III Experimental Unmanned Vehicle (XUV) in the form of Vehicle echelon plans. The point where the Vehicle echelon plan intersected the Subsystem echelon planning horizon was chosen as a goal for the Subsystem echelon path planner. The Primitive echelon BG process used a “Pure Pursuit” algorithm to generate commands to the Servo echelon, where position servo algorithms were implemented to control hydraulic steering actuators and hydrostatic wheel motors.

At each echelon, BG processes (i.e., Planner and Executors) decompose higher echelon commands into lower echelon actions. The Executor at each echelon executes the plan generated by the Planner. Meanwhile, the Planner is replanning based on an updated world state in the KD.

Each planner uses a world model simulator that is appropriate for the problems encountered within the node at its echelon. The Planners and Executors operate asynchronously. At each echelon, the Planner generates a new plan and the Executor outputs new commands to subordinates on the order of ten times within each planning horizon. At each lower echelon, the planning horizon shrinks by a factor of about ten. The relative timing relationships between echelons are designed to facilitate control stability and smooth transitions among hierarchical control levels.

In the world model KD of each echelon above the servo, there exists a map with range and resolution that is appropriate for path planning at that echelon. At each echelon, there are provisions for symbolic data structures and segmented images with labeled regions that describe entities, events, and situations that are relevant to decisions that must be made at that level. These data structures are populated by SP modules that extract information from the sensory data stream that is needed to keep the world model KD current and accurate.

4D/RCS is designed to accommodate a variety of sensors that include a LADAR, stereo CCD cameras, stereo FLIRs, color CCD cameras of different resolutions, vegetation penetrating radar, GPS (Global Positioning System), an inertial navigation package, actuator feedback sensors, and a variety of internal sensors for measuring parameters such as engine temperature, speed, vibration, oil pressure, and fuel level.

The Servo echelon has no map representation. It deals with actuator kinematics and dynamics and reacts to sensory feedback from actuator sensors. The Primitive echelon uses accelerometer data to control vehicle dynamics and prevent roll-over during high speed driving. The Primitive echelon map has range of 5 m with resolution of 4 cm. This enables the vehicle to perform precise maneuvers such as parking or moving through narrow passages, and to make small path corrections to avoid bumps and ruts. The Subsystem echelon map has range of 50 m with resolution of 40 cm. This map is used to plan about 5 s into the future to find a path that avoids obstacles and provides a smooth and efficient ride. The Vehicle echelon map has a range of 500 m with resolution of 4 m. This map is used to plan paths about 1 min into the future taking into account terrain features such as roads, bushes, gullies, or tree lines. The Section echelon map (implemented in Demo III on the Human-Robot Interface computer) has a range of more than 2 km with resolution of about 30 m. This map is used to plan more than 10 min into the future to accomplish tactical behaviors. Higher echelon maps can be used to plan platoon, company, and battalion missions lasting about 1 h, 5 h, and 24 h respectively. These are derived from military maps and intelligence provided by the digital battlefield database.

At all echelons, 4D/RCS planners are designed to generate new plans well before current plans become obsolete. Thus, action always takes place in the context of a recent plan, and feedback through the executors closes reactive control loops using recently selected control parameters. To meet the demands of dynamic battlefield environments, the 4D/RCS architecture specifies that replanning should occur within about one-tenth of the planning horizon at each echelon.

At all echelons Executors are designed with control cycles to react to sensory feedback more quickly than the replanning interval. The Executors monitor feedback from the lower echelons on every control cycle. Whenever an Executor senses an error between its output CommandGoal and the predicted state at the GoalTime (generated by the subordinate BG Planner), it may react by modifying the commanded action so as to cope with that error. This closes a feedback loop through the Executor at that echelon within a specified reaction latency.

The most recent implementation of the 4D/RCS architecture resides on the ARL/NIST HMMWV mobility testbed. On this vehicle, both feedforward and feedback algorithms have been implemented at the Servo echelons to provide precise steering and speed control, and at the Primitive echelon to generate dynamically safe maneuvers. Path planning at the Subsystem echelon provides obstacle avoidance and gaze control for mobility sensors. The most recent application of the 4D/RCS methodology addresses issues at the Vehicle, Section, and Platoon echelons where tactical behaviors for missions such as route reconnaissance are being developed. The Vehicle echelon coordinates all activities on the vehicle, including gaze control for distant vision, communications with the C4ISR network, and mission packages such as weapons and countermine sensors.

IV. TACTICAL BEHAVIORS

The development of tactical behaviors within the 4D/RCS framework begins with the analysis of scenarios in terms of the mission, the tasks that must be performed to accomplish the mission, the environment in which the mission takes place, and the tactics, techniques, and procedures that will be used. [5]

Efforts to embed tactical behaviors in the 4D/RCS architecture have focused on the analysis of scenarios related to the execution of a tactical road march to an assembly area by a light cavalry troop. Activities at the troop commander echelon consist of a number of planning activities such as identifying information to be gathered, defining the organization of the march column, and specifying the formation and movement technique. Within the tactical road march mission, attention has focused on the subtask of route reconnaissance by the Scout Platoon.

It is the responsibility of the scout platoon leader to organize the platoon into sections of vehicles, to assign each section leader responsibility for reconnaissance of different regions along the route, and to select procedures for maintaining security. It is the responsibility of each section leader to evaluate the environment, to select control points for tactical movement of each vehicle in the section, and to coordinate vehicle movements to provide overwatch security. It is the responsibility of each vehicle commander to employ sensors to scan the terrain and analyze the local situation in terms of mission goals, security, stealth, and traversability, and to communicate reports back to the section commander.

Within each vehicle there are subsystems that support reconnaissance, communication, mission packages (such as weapons), and mobility (the driver.) The vehicle driver is responsible for controlling speed and direction that are safe and effective for negotiating the terrain in the immediate vicinity of the vehicle, while reaching control points specified by the vehicle commander in a stealthy and timely manner. Each subsystem consists of actuator groups that provide coordinated behaviors for gaze control and vehicle speed and heading. Finally, at the lowest echelon in the hierarchy, control signals are generated for steering, throttle, and braking systems to achieve the speeds and directions commanded by the driver. The result of this task decomposition process is that each vehicle selects and pursues its own real-time path through the environment.

If some environmental obstruction, such as a water obstacle, constrains the vehicle from reaching the control points laid out by the section leader, the vehicle should respond by reconnoitering the obstacle, moving to a secure point, and reporting the situation to its section leader. If the obstruction is large enough to affect the operation of the entire section (e.g. the water obstacle stretches across the entire area to which the section is assigned), then the section leader would be responsible for generating a new plan for coordinating his vehicles to do reconnaissance and to take up secure positions. The section leader would also report the new situation to the platoon leader.

Detailed analysis of a set of scenarios such as described above provide deep insight into the sensory processing algorithms required to detect and classify objects, events, and situations; the set of knowledge, skills, abilities required to perform the tasks; the planning and reasoning algorithms required to adapt to unexpected events; and the command responsibilities, rules of engagement, and reporting procedures that are required at each echelon of the organizational structure for successful mission execution.

V. THE 4D/RCS METHODOLOGY

The 4D/RCS methodology [6] uses task decomposition as the primary means of understanding the knowledge required for intelligent control. This approach can be summarized in six steps as shown in Figure 4.

The first step consists of knowledge “mining” activities to retrieve knowledge from textbooks, training manuals, and subject matter experts (SMEs). Typically, information from textbooks and manuals “bottoms-out” at the level of tasks that can be performed by trained human warriors. Typically, extensive question and answer sessions with SMEs (often while reenacting scenarios in the field), are required to understand how humans decompose tasks further into activities that control actuators and sensors that produce physical action in the world. At each echelon in this task decomposition process, higher-level commands from upper echelons must be decomposable into a series of lower-level commands that can be further decomposed until bottom level actuator commands are reached. The output of step 1 is a set of tasks that can be executed by intelligent control modules (either human or machine) at each echelon of the command and control hierarchy.

The second step is to map the task decomposition hierarchy developed under step 1 onto an organizational hierarchy of intelligent control modules that are capable of performing the required task decomposition at each level. This step requires a formal structuring of all of the subtask activities as well as definition of a control execution structure including all the timing considerations, messages, and communication mechanisms that are required to coordinate the activities of all of the control modules. 4D/RCS uses a communication system called Neutral Messaging Language (NML) for intermodule communication. [7, 8]

The third step, defines the rules and procedures whereby each task command is decomposed into subcommands to subordinate modules. Rules and procedures for each command are clustered into state-tables that can be executed by extended finite state machines (EFSMs.) The sequence of outputs generated by the EFSMs consist of commands sent to control modules at lower echelons in the Command and Control (C2) hierarchy.

The fourth step identifies all of the world states whose combinations and patterns create the situations that cause the EFSMs to transition from one state to the next. These are represented in a distributed knowledge database that is maintained by the WM process in each module. This knowledge database defines the set of situations that the SP and WM processes must be able to detect and classify.

The fifth step identifies all of the entities, events and relationships that are required to define the situations identified in step 3.

The last step defines the set of attributes that must be sensed in the world to identify and classify the entities, events, and situations identified in steps 4 and 5. This includes the spatial and temporal resolution of sensors, the algorithms required to process the sensory data, e.g., to filter data, focus attention, segment images and signals, compute attributes, and classify segmented regions in the sensory data. It also includes the use of top-down information such as task goals and expectations based on knowledge of the situational context. This information can be used to establish specifications for the sensors and processing algorithms for each subtask activity.

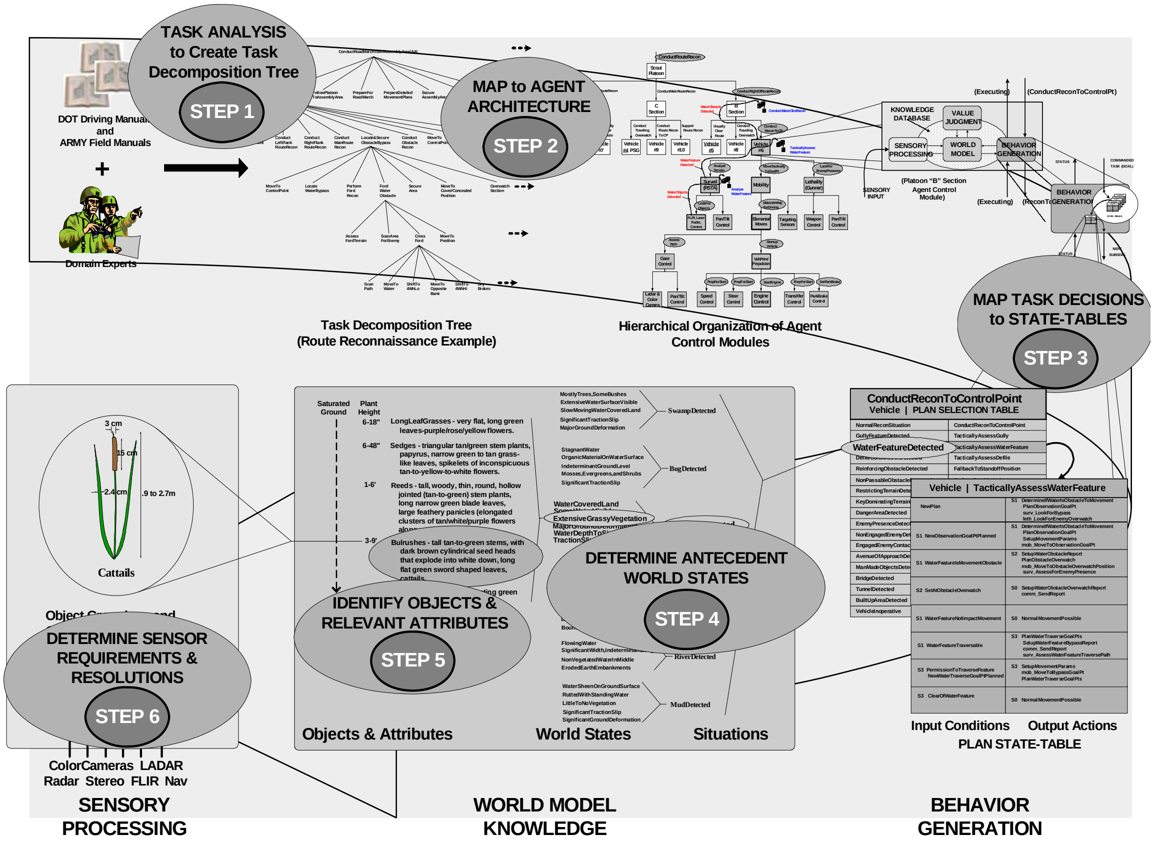

Figure 4. A summary of the 4D/RCS methodology for developing task knowledge.

Output of the 4D/RCS methodology is a set of information requirements for tasks (step 1), computational modules (step 2), rules to accomplish tasks (step 3), situations and their antecedent world states (step 4), environmental entities/objects (step 5), and requirements for sensors and sensory processing algorithms (step 6). More details about this methodology can be found in [6].

Once the complete set of knowledge required to perform a scout platoon route reconnaissance mission has been formalized and analyzed by the above process, an adaptation of the tactics can be done to include the use of unmanned elements in coordination with the manned elements.

VI. SUMMARY AND CONCLUSIONS

We have described the 4D/RCS reference model architecture and design methodology that has been developed over the past 15 years for the control of intelligent systems, from the servo level up to and including heterogeneous groups of autonomous vehicles. The capabilities of 4D/RCS have been achieved as a result of consistent long term funding from the Army Research Laboratory under the direction of Mr. Charles Shoemaker. 4D/RCS was designed specifically for the ARL Demo III Experimental Unmanned Vehicle program. It has subsequently been adopted by the Army Tank and Automotive Research and Development Center (TARDEC) for their Vetronics Technology Integration program, and by General Dynamics Robotic Systems (GDRS) for the Autonomous Navigation System that will be used on all ground vehicles (manned and unmanned) in the Army Future Combat System. The 4D/RCS design methodology is currently being used by GDRS, ARL, and the Ft. Knox Unit of Action Maneuver Battle Lab for specifying tactical behaviors in sufficient detail so that they can be implemented in unmanned vehicles (both ground and air) for the Army Future Combat System.

ACKNOWLEDGEMENTS

This work was supported in part by the Army Research Lab program in unmanned vehicles, Mr. Charles Shoemaker, Program Manager.

REFERENCES

1. J. ALBUS and et.al. 2002 “4D/RCS Version 2.0: A Reference Model Architecture for Unmanned Vehicle Systems,” *NISTIR 6910*, National Institute of Standards and Technology, Gaithersburg, MD.
2. J. ALBUS and A. MEYSTEL, *Engineering of Mind: An Introduction to the Science of Intelligent Systems*. New York, NY: John Wiley & Sons, Inc., 2001
3. E. DICKMANNS, “An Expectation-based, Multi-focal, Saccadic (EMS) Vision System for Vehicle Guidance,” *Proceedings 9-th Int. Symp. on Robotics Research (ISRR’99)* Salt Lade City, October, 1999.
4. C. SHOEMAKER SPIE paper
5. J. ALBUS, A. BARBERA, S. BALAKIRSKY, and H. SCOTT, “Route Reconnaissance with UAVs and UGVs in the FCS Unit of Action,” *Proceedings of the SPIE AeroSense Session on Unmanned Ground Vehicle Technology,* Orlando, FL, 2005.
6. A. BARBERA., J. ALBUS, E. MESSINA, C.SCHLENOFF and J. HORST., "How Task Analysis Can Be Used to Derive and Organize the Knowledge For the Control of Autonomous Vehicles," *The Robotics and Autonomous Systems Journal: Special Issue on the 2004 AAAI Knowledge Representation and Ontologies for Autonomous Systems Spring Symposium*, 2004.
7. V. GAZI., M.L. MOORE, K.M. PASSINO, W.P. SHACKLEFORD, F.M. PROCTOR, and J. ALBUS, *The RCS Handbook: Tools for Real Time Control Systems Software Development,* John Wiley & Sons, NY, 2001.
8. http://www.isd.mel.nist.gov/projects/rcs/