Progress and Prospects for Collaborative Tactical Behaviors for Autonomous Ground and Air Vehicles

by

James Albus, Anthony Barbera, Craig Schlenoff

National Institute of Standards and Technology

Phone 301-975-3418   FAX 301-990-9688

james.albus@nist.gov, anthony.barbera@nist.gov, craig.schlenoff@nist.gov

http://www.isd.mel.nist.gov/

Abstract

Tactical behaviors by individuals require situational awareness, including knowledge of the environment, self-knowledge, knowledge of the rules of engagement, knowledge of the disposition, intent, and capabilities of the enemy, and knowledge of appropriate tactics, techniques, and procedures. Collaborative tactical behaviors require additional knowledge of the state, capabilities, and intent of commanders, peers, and subordinates. Recent results of research with the 4D/RCS reference model architecture suggest how all of these kinds of knowledge can be represented in a real-time dynamic world model, how this knowledge can be updated in real-time from processed sensor data or from battlefield information systems. This paper describes an architecture and design methodology that support the sharing of procedural and declarative knowledge by teams of autonomous ground and air vehicles to generate collaborative tactical behaviors that accomplish complex mission assignments.

1. Introduction

Unmanned vehicle technology has progressed to the point where militarily useful tactical behaviors by autonomous systems are becoming feasible. Since 2001, the Army Research Lab Demo III program has demonstrated that experimental unmanned vehicles can autonomously avoid stationary obstacles and maneuver at speeds of 10 km per hour over a variety of terrain types, including woods, rolling fields of tall grass and weeds, dirt roads, erosion features, and urban environments containing buildings, telephone poles, cars, and piles of rubble (1). Technology for following paved roads and avoiding collisions with other vehicles moving in the same direction has also been demonstrated at the German Military University in Munich. Autonomous vehicle speeds of 180 km per hour have been achieved in traffic on the German Autobaun (2). Current research is focused on coping with on-coming traffic on two-lane roads, negotiating intersections, and reacting to traffic signs and signals. Work has now begun on developing collaborative military tactical behaviors between Uninhabited Air Vehicles (UAVs), Unmanned Ground Vehicles (UGVs), and manned vehicles for missions such as route reconnaissance, and detecting/neutralizing improvised explosive devices. In this paper, we will describe the 4D/RCS reference model architecture and design methodology that was developed for the Demo III program, and show how it has been applied to date for on-road autonomous driving and tactical behaviors. We will also suggest how it could support collaborative tactical behaviors for autonomous ground and air vehicles.

2. Background of 4D/RCS

4D/RCS is both a reference model architecture and a design methodology (3, 4, 5). The 4D/RCS 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). At typical 4D/RCS node is shown in Figure 1. (Solid lines indicate normal data pathways. Dotted lines indicate channels by which an operator can peek at data or insert control commands whenever desired.) Each 4D/RCS node corresponds to an organizational unit consisting of a supervisor agent and a collection of subordinate agents. A supervisor agent at one echelon of the hierarchy is typically a subordinate agent in a node at the next higher level. A subordinate agent at one echelon is typically a supervisor agent in a node at the next lower level.

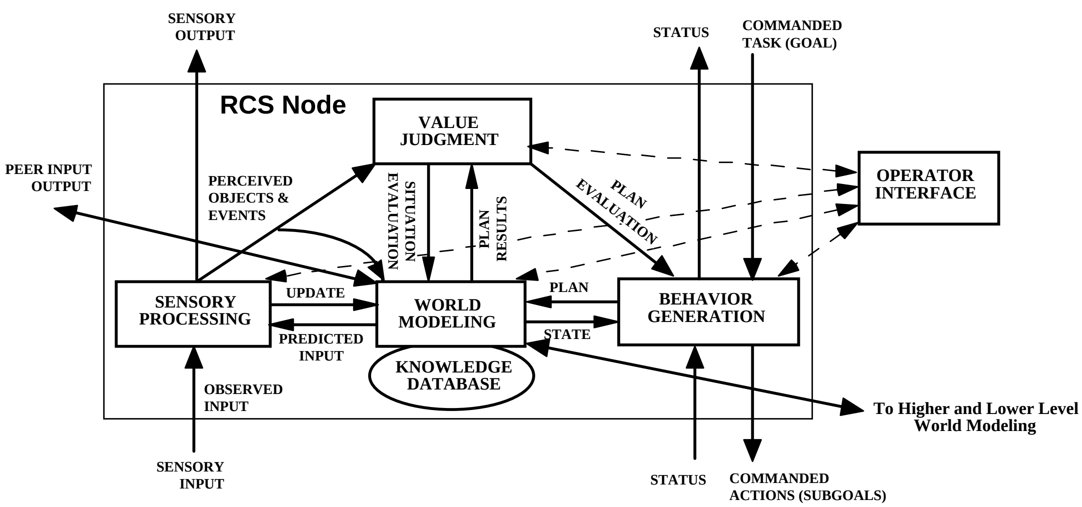

Figure 1: 4D/RCS Node

Throughout the hierarchy, interaction within the nodes between SP, WM, VJ, BG, and KD give rise to perception, cognition, and reasoning. In low-level nodes, representations of space and time are short-range and high-resolution. In high-level nodes, distance and time are long-range and low-resolution. This enables high-precision fast-acting response in low-level nodes, while long-range plans and abstract concepts are being simultaneously formulated in high-level nodes. This hierarchical approach also helps to manage computational complexity. Any node has finite number of duties and responsibilities, with a limited range and resolution of spatial and temporal knowledge to be represented.

4D/RCS closes feedback loops through nodes at every level. SP processes focus attention (i.e., window regions of space or time), perform grouping operations (i.e., segment regions into entities), compute entity attributes, estimate entity state, and assign entities to classes at every level. WM processes maintain a rich and dynamic database of knowledge about the world in the form of images, maps, entities, events, and relationships at every level. Other WM processes use that knowledge to generate estimates and predictions that support perception, reasoning, and planning at every level. VJ processes assign worth and importance to objects and events, compute confidence levels for variables in the knowledge database, and evaluate the results of actions, and the anticipated results of hypothesized plans.

The 4D/RCS reference model architecture has been successfully implemented in the Demo III Experimental Unmanned Vehicle program for autonomous mobility in complex (but so far stationary) environments (6). It has been adopted by General Dynamics Robotic Systems[1] for the Autonomous Navigation System for the Army Future

[1] The mention of a company or product by name does not imply endorsement by the National Institute of Standards and Technology.

Combat System program. Current NIST research is directed toward using 4D/RCS to implement tactical behaviors by groups of robots in dynamic environments.

3. RCS Design Methodology

The fundamental premise of the RCS Design Methodology is that at each point in time, the task state (i.e., where the system is, where it is going, what it is doing, what its goal is, and what the constraints are) collectively define the requirements for all of the knowledge in the knowledge database (both procedural and declarative), and specifies the support processing required to acquire and maintain the knowledge database. In particular, the task state determines what needs to be sensed, what world objects, events, and situations need to be analyzed, what plans need to be generated, and what task knowledge is required to do so (7,8.) An example of the RCS methodology for designing a control system for a tactical behavior such as route reconnaissance is shown in Figure 2.

The RCS design methodology begins with an in-depth analysis of the task or mission the system is intended to perform. This is followed by the encoding of task knowledge (i.e., procedural knowledge) in the form of a task decomposition tree (e.g., AND/OR graphs) that represents the decomposition of tasks into sequences of simpler and simpler subtasks (as shown in Step 1 of Figure 2.) This task decomposition framework is then mapped onto a hierarchy of organizational units that have the knowledge, skills, and abilities to perform the required task decompositions (step 2 of Figure 2.) The procedures or plans along with necessary state transition events are defined for each task (step 3 of Figure 2.) The decomposition of each state transition event into finer grained events is then determined (step 4 of Figure 2.) The world model variables (i.e., declarative knowledge) required for each event are defined (step 5 of Figure 2) as well as the precise functions mapping variables to events. Finally, the sensors and sensory processing capabilities required to generate and maintain the values of each world model variable are specified (5) (step 6 of Figure 2.)

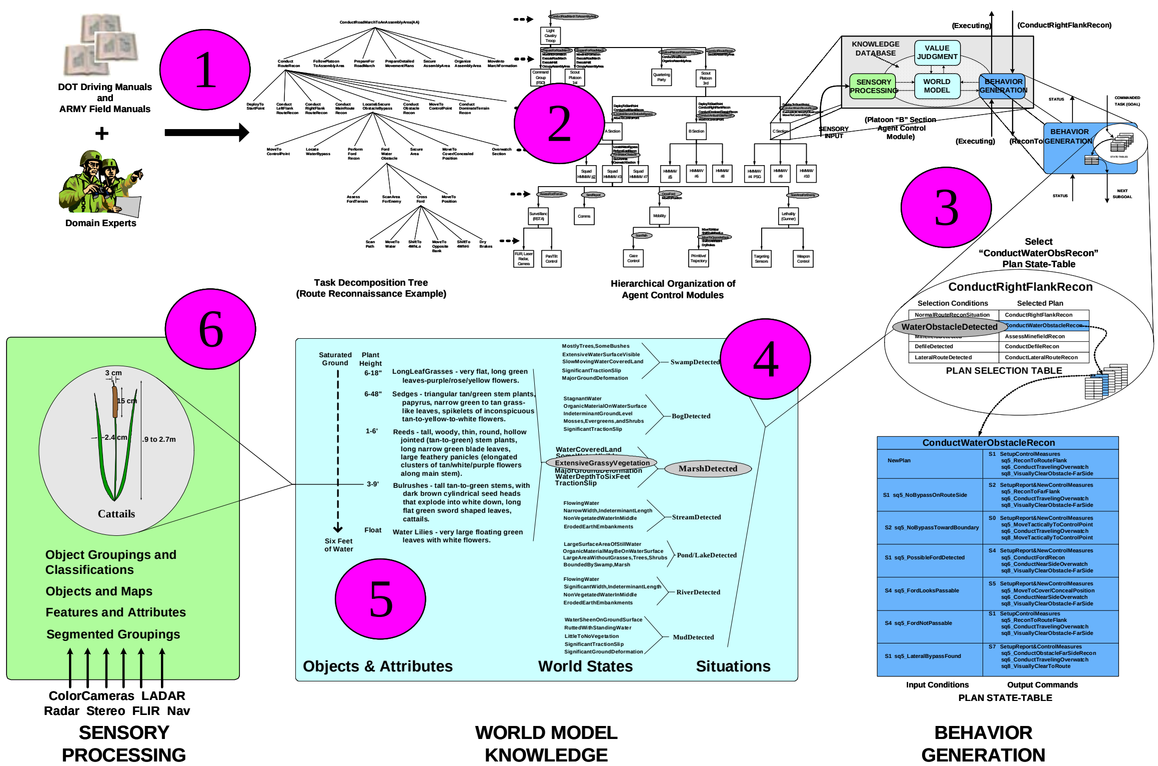

Figure 2. The six steps of the RCS design methodology for knowledge acquisition and representation

Step 1 consists of an intensive analysis of domain knowledge derived from training manuals and subject matter experts. Scenarios are developed and analyzed for each task and subtask. The result of this step is a structuring of procedural knowledge into a task decomposition tree with simpler and simpler tasks at each echelon. At each echelon, a vocabulary of commands (i.e., action verbs with goal states, parameters, and constraints) is defined to evoke task behavior at each echelon.

Step 2 defines a hierarchical structure of organizational units that will execute the commands defined in step 1. For each unit, duties and responsibilities in response to each command are specified. This is analogous to establishing a work breakdown structure for a development project, or defining an organizational structure for a business or military operation.

Step 3 specifies the processing that is triggered within each unit upon receipt of an input command. For each input command, a state-graph (or state-table, or extended finite state automaton) is defined that provides a plan (or procedure for making a plan) for accomplishing the commanded task. The input command selects (or causes to be generated) an appropriate behavior (which may be encoded as a state-table), the execution of which generates a series of output commands to units at the next lower echelon. The result of step 3 is that each organizational unit has, for each input command, a state-table of production rules that identify all the task branching conditions and specify the corresponding state transition and output command parameters.

Step 4 analyzes each of the situations defined in step 3 to reveal dependencies on world states and situations. This step identifies the detailed relationships between entities, events, and states of the world that cause each state or situation to be true.

Step 5 identifies and names all of the world model entities and events along with their attributes and relationships that are relevant to detecting the world states and situations.

Step 6 uses the context of world states and situations to establish the distances and, therefore, the resolutions at which the relevant entities, events, and situations must be measured and recognized by the sensory processing component. This establishes a set of requirements and/or specifications for the sensor system to support each subtask activity.

4. Representing Knowledge in 4D/RCS

The 4D/RCS architecture is designed to accommodate multiple types of representation formalisms, and provide an elegant way to integrate these formalisms into a common unifying framework. This section will describe the types of knowledge representations that have been researched and/or implemented within the 4D/RCS architecture for autonomous driving and the mechanisms that have been deployed to integrate them.

The hierarchical structure of 4D/RCS supports knowledge representation with different range and resolution, and different levels of abstraction, at the various echelons of control. It should be noted, however, that the 4D/RCS reference model actually contains three different hierarchies:

1. a task hierarchy consisting of a chain of command with echelons of control;
2. a hierarchy of range and resolution of signals, images, and maps in space and time;
3. a hierarchy of abstraction in representation of entities and events.

Echelons of control are defined by decomposition of tasks into subtasks and the assignment of task skills and responsibilities to organizational units in a chain of command. Range and resolution of signals, images, and maps are defined by sampling interval and field of regard over space and time (e.g., pixel size & field of view in images, scale and size of maps, and sampling frequency of signals.) Levels of abstraction are defined by grouping and segmentation algorithms that operate on the geometry of entities (e.g., points, lines, vertices, surfaces, objects, groups) and the duration of events (e.g., milliseconds, seconds, minutes, hours, days.) These three hierarchies are related, but not congruent. For example, the range and resolution of maps are related to echelons of control by speed and size of the system being controlled. Resolution of images is related to spatial dimension by magnification. Resolution of maps is related to spatial dimension by scale. Pixels in images are related to pixels on maps by transformation of coordinates.

The RCS methodology begins with the task decomposition hierarchy that defines echelons of control. Task timing and system speed and size then determine range and resolution of images and maps. Levels of abstraction are determined by the logical requirements of task decomposition. Different system requirements will produce different relationships between these three hierarchical representations.

Typically, knowledge in 4D/RCS nodes at the lowest echelon of the control hierarchy consist of signals, images, and state variables, including vehicle position, orientation, velocity, and acceleration; along with actuator positions, velocities, and forces; pressure sensor readings, position of switches, and gearshift settings. Knowledge in nodes at the second echelon and above consists of map-based information, with decreasing resolution and increasing spatial extent at each higher echelon in the hierarchy. Maps are used to represent the size, shape, location, surface orientation, and roughness of terrain features and regions of interest. Knowledge in nodes at the third echelon and above contain both map-based representations and abstract data structures that represent named entities such as road edges and obstacle surfaces, along with their attributes and pointers that represent spatial and temporal relationships and class membership. Image and map representations are linked to abstract data structures by pointers that represent relationships such as “is_a” and “belongs_to.” Back pointers indicate where abstract entities are located in the map and/or image representations. Higher echelons represent information about the location, motion, and attributes of objects such as other vehicles, roads, intersections, traffic signals, landmarks, and terrain features such as buildings, roads, woods, fields, streams, fences, and ponds. The upper level echelons represent knowledge about groups of objects such as groups of people, groups of buildings, and road networks. Group attributes such as size, shape, density, are computed over the group. At each echelon, pointers define relationships between entities and events in situations.

Within each node, knowledge is stored within a Knowledge Database (KD) consisting of data structures that contain the static and dynamic information that collectively form a model of the world. Each node contains knowledge with the range, resolution, and level of abstraction required to support the behavior generation, sensory processing, and value judgment processes within that node. This includes a best estimate of the current state of the world relevant to the current task assigned to that node, plus parameters that define how the world state can be expected to evolve in the future under a variety of circumstances.

Figure 3 shows the many different types of knowledge representation formalisms that are currently being implemented within the 4D/RCS architecture to support autonomous driving (9). These formalisms range from iconic to symbolic and from procedural to declarative. Knowledge is captured in formalisms and at levels of abstraction that are suitable for the way that it is expected to be used. Different knowledge representation techniques offer different advantages, and 4D/RCS is designed in such a way as to combine the strengths of all of these techniques into a common unifying architecture in order to exploit the advantages of each.

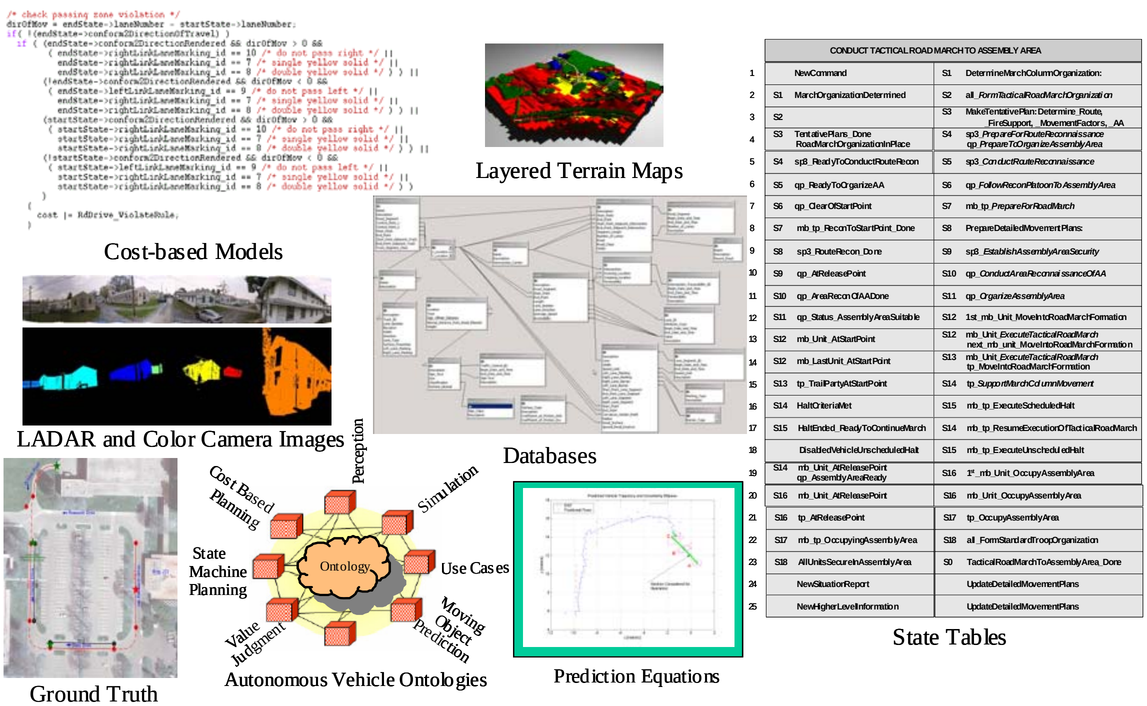

Figure 3: Knowledge Representations in 4D/RCS

5. Experimental Results

Experimental validation of the 4D/RCS architecture has been provided by the performance of the Demo III experimental unmanned ground vehicles (XUVs) in an extended series of demonstrations and field tests during the winter of 2002-2003 (1).

The XUVs were equipped with an inertial reference system, a commercial grade GPS receiver (accurate to about +/- 20 m), a LADAR camera with a frame rate of 10 frames per second, and a variety of internal sensors. The LADAR had a field of view 90^o wide and 20^o high with resolution of about 0.5^o per pixel. It was mounted on a pan/tilt head that enabled it to look in the direction that it planned to drive. The LADAR was able to detect the ground out to a range of about 20 m, and detect vertical surfaces (such as trees) out to a range of about 60 m. Routes for XUV missions were laid out on a terrain map by trained Army scouts, and given to the XUVs in terms of GPS waypoints spaced more than 50 m apart.

The XUVs operated completely autonomously until they got into trouble and called for help. Typical reasons for calling for help were the XUV was unable to proceed because of some terrain condition or obstacle (such as soft sand on a steep slope, or dense woods), and was unable to find an acceptable path plan after several attempts at backing up and heading a different direction. At such a point, an operator was called in to teleoperate the vehicle out of difficulty. During these operations, data was collected on the cause of the difficulty, the type of operator intervention required to extract the XUV, the time required before the XUV could be returned to autonomous mode, and the work load on the operator.

During three major experiments designed to determine the technology readiness of autonomous driving, the Demo III experimental unmanned vehicles were driven a total of 550 km, over rough terrain: 1) in the desert; 2) in the woods, through rolling fields of weeds and tall grass, and on dirt roads and trails; and 3) through an urban environment with narrow streets cluttered with parked cars, dumpsters, culverts, telephone poles, and manikins. Tests were conducted under various conditions including night, day, clear weather, rain, and falling snow. The unmanned vehicles operated over 90 % of both time and distance without any operator assistance. A detailed report of these experiments has been published (1), along with high-resolution ground truth data describing the terrain where the XUVs experienced difficulties (7).

It should be noted that the Demo III tests were performed in environments devoid of moving objects such as on-coming traffic, pedestrians, or other vehicles. The inclusion of moving objects in the world model, and the development of perception, world modeling, and planning algorithms for operating in the presence of moving objects is a topic of current research.

6. Conclusion

In this paper, we have described how 4D/RCS supports multiple types of representations, ranging from iconic to symbolic and from declarative to procedural, and provided brief examples of how each of these representations are used in the context of autonomous driving and tactical behaviors. Recent results suggest that this same approach would lend itself well to collaborative tactical behavior for autonomous ground and air vehicles, specifically in facilitating the explicit capture and sharing of information for joint operations.

Research in our lab is now focused on two aspects of autonomous vehicle control:

1) autonomous driving on normal roads and streets, e.g., driving on country roads and city streets with on-coming traffic, negotiating intersections with traffic signals and pedestrians, and maneuvering in and out of parking spaces

2) autonomous tactical behaviors for teams of real and virtual autonomous military ground and air vehicles, e.g., controlling the behavior of a platoon of scout vehicles consisting of seven manned and three unmanned ground vehicles, and three unmanned air vehicles cooperating in the performance of a route reconnaissance mission prior to a troop echelon road march.

Experience on the Demo III program has demonstrated that the 4D/RCS architecture provides an excellent framework for building intelligent systems that perform useful tactical behaviors under combat conditions.

6. References

1. Camden, R., Bodt, B., Schipani, S., Bornstein, J., Phelps, R., Runyon, T., French, F., and Shoemaker, C., "Autonomous Mobility Technology Assessment Interim Report," *Army Research Laboratory (ARL-MR-565)*, 2003
2. Dickmanns, E. D., "An Expectation-Based Multi-Focal Saccadic (EMS) Vision System for Vehicle Guidance," *Proceedings of the 9th International Symposium on Robotics Research (ISRR'99)*, Salt Lake City, 1999
3. Albus, J. and et.al., "4D/RCS Version 2.0: A Reference Model Architecture for Unmanned Vehicle Systems," NISTIR 6910, National Institute of Standards and Technology, Gaithersburg, MD, 2002

4. Albus, J. and Meystel, A., *Engineering of Mind*, John Wiley & Sons, Inc. 2001
5. Albus, J. and Barbera, A. "RCS: A Cognitive Architecture for Intelligent Multi-Agent Systems," *Proceedings of Intelligent Autonomous Vehicles Conference*, Lisbon, Portugal, 2004
6. Shoemaker, C., Bornstein, J., Myers, S., and Brendle, B., "Demo III: Department of Defense testbed for unmanned ground mobility," *SPIE Conference on Unmanned Ground Vehicle Technology, SPIE Vol. 3693*, Orlando, FA, April, 1999
7. Barbera, A., Horst, J., Schlenoff, C., Wallace, E., Aha, D., "Developing World Model Data Specifications as Metrics for Sensory Processing for On-Road Driving Tasks," *Proceedings of the 2003 PerMIS Workshop, NIST Special Publication 990*, Gaithersburg, MD, 2003
8. Barbera, A., Albus, J., Messina, E., Schlenoff, C., Horst, J., "How Task Analysis Can Be Used to Derive and Organize the Knowledge for the Control of Autonomous Vehicles," *Proceedings of the 2004 AAAI Spring Symposium on Knowledge Representation and Ontologies for Autonomous Systems*, Palo Alto, CA, 2004
9. Witzgall, C., Cheok, G.S., Gilsinn, D.E., "Terrain Characterization from Ground-Based LADAR," *Proceedings of PerMIS '03 Workshop*, National Institute of Standards and Technology, Gaithersburg, MD, 2003