

Using 4D/RCS to Address AI Knowledge Integration

Craig Schlenoff, Jim Albus, Elena Messina, Anthony J. Barbera, Raj Madhavan, and Stephen Balakirsky

■ In this article, we show how 4D/RCS incorporates and integrates multiple types of disparate knowledge representation techniques into a common, unifying architecture. The 4D/RCS architecture is based on the supposition that 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. In the context of applying the architecture to the control of autonomous vehicles, we describe the procedural and declarative types of knowledge that have been developed and applied and the value that each brings to achieving the ultimate goal of autonomous navigation. We also look at symbolic versus iconic knowledge representation and show how 4D/RCS accommodates both of these types of representations and uses the strengths of each to strive towards achieving human-level intelligence in autonomous systems.

During the past century, the neurosciences have provided deep insights into the anatomical, physiological, chemical, and computational bases of cognition. Neuroanatomy has described the structure and function of the basic computational element of the brain—the neuron—and produced extensive maps of the computational modules and interconnecting data flow pathways making up

the anatomy of the brain. Behavioral psychology provides information about stimulus-response behavior and instrumental conditioning. Cognitive psychology is exploring how the brain represents knowledge; how it perceives objects, events, situations, and relationships; how it analyzes the past and plans for the future; and how it selects and controls behavior that satisfies desires and achieves goals

Over the last five decades, the invention of the electronic computer has brought rapid advances in computational power, making it feasible to launch serious attempts at building intelligent systems. Artificial intelligence and robotics have produced significant results in planning, problem solving, rule-based reasoning, image analysis, and speech understanding. Autonomous vehicle research has produced advances in real-time sensory processing, world modeling, navigation, path planning, and obstacle avoidance. Research in industrial automation and process control has produced hierarchical control systems, distributed databases, and models for representing processes and products. Modern control theory has developed precise understanding of stability, adaptability, and controllability under various conditions of uncertainty and noise. Progress is rapid in each of the above fields, and there exists an enormous and rapidly growing body of literature in all of these areas.

What is lacking is a widely accepted theoret-

ical architecture that can integrate concepts from all of these different fields into a unified whole. This article describes the 4D/RCS architecture and describes how it has been implemented to leverage and integrate multiple different types of knowledge representation in the domain of autonomous vehicle navigation.

The “Related Architectures” section gives an overview of existing intelligent architectures. That section is followed with a presentation of the background of 4D/RCS. We then describe, in the “Intelligence in Autonomous Vehicles” section, knowledge that is captured in 4D/RCS as it relates to enabling intelligence in autonomous vehicles. Finally, we conclude with sections describing the results and concluding remarks.

Related Architectures

One of the earliest architectures was the ACT architecture (Anderson 1983). ACT grew out of research on human memory. Over the years, ACT has evolved into ACT* and more recently, ACT-R. ACT-R is being used in several research projects in an Advanced Decision Architectures Collaborative Technology Alliance for the U.S. Army (Gonzalez 2003). ACT-R is also being used by thousands of schools across the country as an algebra tutor—an instructional system that supports learning by doing. Another well-known and widely used architecture is Soar (Laird, Newell, and Rosenbloom 1987). Soar grew out of research on human problem solving and has been used for many academic and military research projects in problem solving, language understanding, computational linguistics, theorem proving, and cognitive modeling.

Other architectures include Prodigy, Icarus, the improved performance research integration tool (IMPRINT), executive-process interactive control (EPIC), and 4D/RCS (4D refers to three dimensions of space and one dimension of time, and RCS stands for real-time control systems). Like Soar, Prodigy uses search through a problem space to achieve goals cast as first-order expressions (Minton 1990). Icarus is a reactive architecture that encodes knowledge as reactive skills (Shapiro and Langley 1999). IMPRINT is a task description language designed for the U.S. Army to capture the procedural specification of tactical behavior scenarios (Archer and Adkins 1999). It contains a dynamic, stochastic, discrete-event network modeling tool designed to help assess the interaction of soldier and system performance throughout the system lifecycle—from concept and design through field testing and system

upgrades. IMPRINT has been integrated with ACT-R to model military behaviors (Archer et al. 2003). EPIC is an architecture that models the detailed timing of human perceptual, cognitive, and motor activity, including the input/output characteristics of the nervous system connecting the higher-level cognitive functions to the external world (Kieras and Meyer 1997). The 4D/RCS architecture is a control system architecture inspired by a theory of cerebellar function (Albus 1971). It models the brain as a hierarchy of goal-directed sensory-interactive intelligent control processes that theoretically could be implemented by neural nets, finite state automata, cost-guided search, or production rules (Albus 1981).

The 4D/RCS architecture is similar to other cognitive architectures in that it represents procedural knowledge in terms of production rules and represents declarative knowledge in abstract data structures such as frames, classes, and semantic nets. It differs from other cognitive architectures in that it also includes signals, images, and maps in its knowledge database, and maintains a tight real-time coupling between iconic and symbolic data structures in its world model. The 4D/RCS architecture is also different in its (1) focus on task decomposition as the fundamental organizing principle; (2) level of specificity in the assignment of duties and responsibilities to agents and units in the behavior-generating hierarchy; and (3) emphasis on controlling real machines in real-world environments.

Background of 4D/RCS

The 4D/RCS architecture evolved from the bottom up as a real-time intelligent control system for real machines operating on real objects in the real world. The first version of RCS was developed as a sensory-interactive goal-directed controller for a laboratory robot (Barbera, Albus, and Fitzgerald). The latter, that is, fundamental element is the control loop, which has a goal, a transition function, a feedback loop, and an action output such as a force, velocity, or position. Over the years, RCS has evolved into an intelligent controller for industrial robots, machine tools, intelligent manufacturing systems, automated general mail facilities, automated stamp distribution systems, automated mining equipment, unmanned underwater vehicles, and unmanned ground vehicles (Albus 1997, Barbera et al. 1984). The most recent version of RCS (4D/RCS) embeds elements of Dickmanns’s (1999) four-dimensional approach to machine vision within the 4D/RCS control architecture. The 4D/RCS architecture

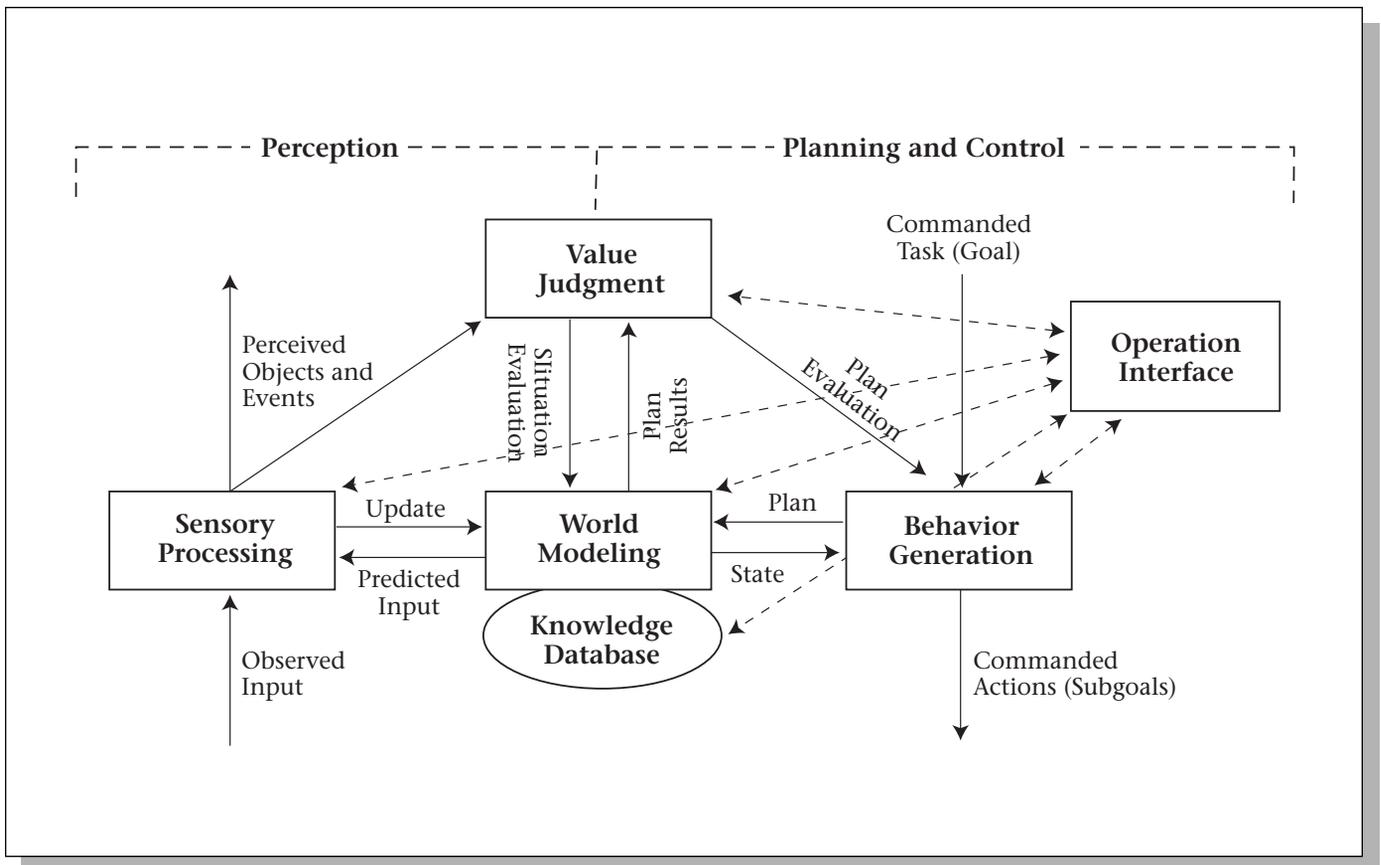


Figure 1. RD/RCS Node.

was designed for the U.S. Army Research Lab AUTONAV and Demo III Experimental Unmanned Vehicle programs and has been adopted by the Army Future Combat System program for Autonomous Navigation Systems (Albus et al. 2002, Albus and Meystel 2001).

The 4D/RCS architecture consists of a multi-layered, multiresolutional 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), as shown in figure 1. Throughout the hierarchy, interaction between SP, WM, VJ, BG, and KD give rise to perception, cognition, and reasoning. At low levels, representations of space and time are short range and high resolution. At high levels, distance and time are long range and low resolution. This enables high-precision fast-action response at low levels, while long-range plans and abstract concepts are being simultaneously formulated at high levels. The hierarchical approach also helps to manage computational complexity.

The 4D/RCS architecture closes feedback

loops at every level, through every node. SP processes focus attention (that is, window regions of space or time), group (that is, 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 anticipated results of hypothesized plans.

Intelligence in Autonomous Vehicles

The 4D/RCS architecture is designed in such a way as to accommodate multiple types of representation formalisms and provide an elegant

Knowledge in the KD includes the system's best estimate of the current state of the world plus parameters that define how the world state can be expected to evolve in the future under a variety of circumstances.

Figure 2 shows the many different types of knowledge representation formalisms that are currently being implemented within the 4D/RCS architecture as applied to autonomous driving. 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. In the following subsections, we describe some of the formalisms depicted, classifying knowledge as either procedural or declarative.

Procedural Knowledge

Procedural knowledge is the knowledge of how to perform tasks. Procedural knowledge is different from other kinds of knowledge, such as declarative knowledge, in that it can be directly applied to a task. Within 4D/RCS, procedural knowledge is primarily used for planning and control purposes. As such, we will describe two planning approaches that are currently being implemented in 4D/RCS and describe the knowledge that underlies each.

Both planning approaches start with the same 4D/RCS methodology for determining the knowledge that needs to be represented to accomplish the planning task. The methodology starts as follows:

The first step involves an intensive analysis of domain knowledge from manuals and subject matter experts, especially using scenarios of particular subtask operations. The output of the effort is a structuring of this knowledge into a task decision tree consisting of simpler and simpler commands (actions/verbs) at simpler and simpler levels of task description.

The second step defines the hierarchical organization of agent control modules that will execute these layers of commands in such a manner as to reasonably accomplish the tasks. This is the same as coming up with a business or military organizational structure of agent control modules (people, soldiers) to accomplish the desired tasks. This step forces a more formal structuring of all of the subtask activities and responsibilities, as well as defining the execution structure.

At this point, the two approaches diverge in

the procedure for determining the types of knowledge necessary to accomplish the planning task. Subsequent steps are described in the following subsections.

State Machine-Based Planning (a). The state machine-based methodology, shown in figure 3, concentrates on the task decomposition as the primary means of understanding the knowledge required for intelligent control. Once the previous two steps are performed, the procedure proceeds as follows:

The third step (3a) clarifies the processing of each agent's input command through the use of rules to identify all of the task branching conditions with their corresponding output commands. Each of these command decompositions at each agent control module will be represented in the form of a state table of ordered production rules (which is an implementation of an extended finite state machine [FSM]). The sequence of simpler output commands required to accomplish the input command and the named situations (branching conditions) that move the state table to the next output command are the primary knowledge represented in this step.

In the fourth step (4a), the above-named situations that are the task branching conditions are defined in great detail in terms of their dependencies on world and task states. This step attempts to define the detailed precursor states of the world that cause a particular situation to be true.

In the fifth step (5a), we identify and name all of the objects and entities together with their particular features and attributes that are relevant to defining the above world states and situations. Current efforts are exploring the use of ontologies and databases to represent this information.

The sixth step (6a) uses the context of the particular task activities to establish the distances and, therefore, the resolutions at which the above objects and entities must be measured and recognized by the sensory processing component. This step establishes a set of requirements and specifications for the sensor system at the level of each separate subtask activity.

Cost-Based Planning Representations (b). The cost-based methodology concentrates on decomposing each of its assigned tasks into an optimal sequence of commands that will be assigned to its subordinates. This is accomplished through the incremental creation and evaluation of a planning graph (Balakirsky 2003). Once again, the first two steps from the procedural knowledge subsection must be performed and are then followed by the following steps:

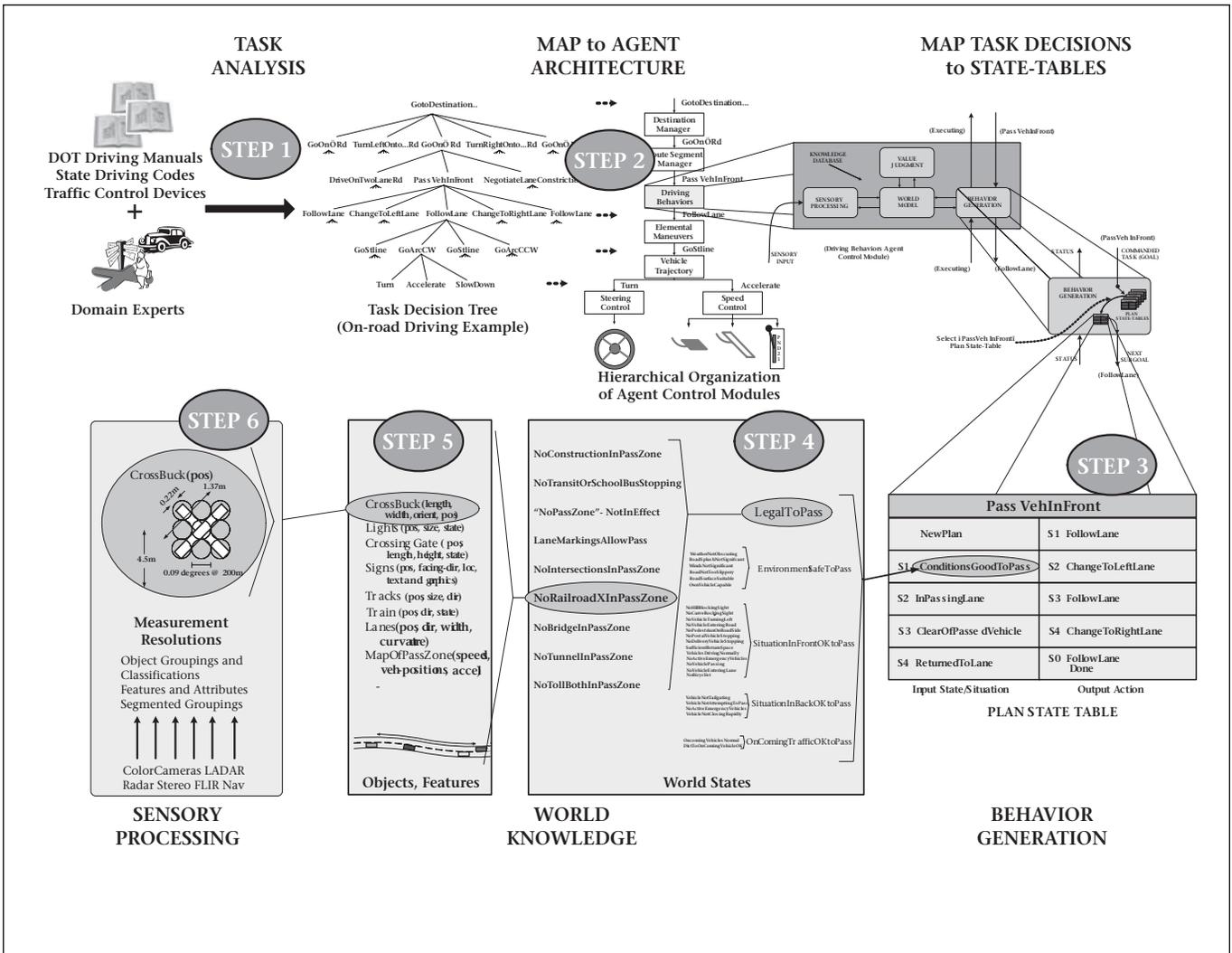


Figure 3. The RCS State Machine-Based Planning Approach.

The third step (3b) develops an action model that delineates how each of the subordinate's commands will affect the system state at the current level of resolution. This allows a simulation system to experiment with various command options in order to obtain the state transitions that are required to fulfill the level's goals.

The fourth step (4b) develops a set of user constraints and objectives that could affect the cost-benefit ratio of performing a given action or occupying a given state. For example, the cost-benefit of running a red light would be substantially different for a casual driver than it would be for a person driving his wife to the hospital to deliver a baby.

Step 5 (5b) examines the potential state variable transitions that have been identified along

with the potential user constraints and objective in order to construct a cost function that will be utilized by the value judgment module during the graph expansion process.

By developing the state transition simulator from step 3b, we are able to incrementally build a planning graph based on potential actions that a subordinate may take. The cost function developed in step 5b may then be used to evaluate the individual arcs of the planning graph in order to control the expansion order and find the cost optimal path through the planning graph.

To represent the knowledge coming out of these methodologies, active efforts have been exploring the development of an ontology to model tactical behaviors. The ontology is based upon the OWL-S specification (Web Ontology

Language-Services).¹ In this context, behaviors are actions that an autonomous vehicle is expected to perform when confronted with a pre-defined situation. The ontology is stored within the 4D/RCS knowledge database, and the behaviors are spawned when situations in the world are determined to be true, as judged by sensor information and the value judgment components. More information about this effort can be found in Schlenoff, Washington, and Barbera (2004).

Declarative Knowledge

Declarative knowledge is represented in a format that may be manipulated, decomposed, and analyzed by reasoners. Unlike procedural knowledge, it does not describe how to perform a given task. Instead, it provides the ability to use knowledge in ways that the system designer did not foresee. Two classes of declarative knowledge captured within 4D/RCS are symbolic knowledge and iconic knowledge. In the follow two subsections, we describe details about these two types of knowledge representations.

Symbolic Knowledge. Symbolic representations provide ways of expressing knowledge and relationships, and of manipulating knowledge, including the ability to address objects by property.

Tying symbolic knowledge back into the spatial representation provides symbol grounding, thereby solving the previously noted problem inherent to purely symbolic knowledge representations. It also provides the valuable ability to identify objects from partial observations and then extrapolate facts or future behaviors from the symbolic knowledge.

Two types of symbolic representations being implemented within 4D/RCS are ontologies and relational databases.

Ontologies represent key concepts, their properties, their relationships, and their rules and constraints within a given domain. Ontologies often focus more on the meaning of concepts than on the terms that are used to represent them. Two efforts have focused on the development of ontologies for autonomous navigation.

A roadway driving ontology that is used to determine whether objects in the environment are potential obstacles to your vehicle has been developed for autonomous driving. The system is composed of an ontology of objects representing “things” that may be encountered in our current environment, in conjunction with rules for estimating the damage that would be incurred by collisions with the different objects in different situations. Automated reasoning is

used to estimate collision damage, and this information is fed to the route planner to help it decide whether to avoid the object. More information about this effort can be found in the paper by Provine et al. (2004).

In addition to ontologies, databases have been developed to house symbolic information. The database that has received the most attention to date is the Road Network Database (Schlenoff et al. 2004). The database includes detailed information about the roadway, such as where the road lies, rules dictating the traversal of intersections, lane markings, road barriers, road surface characteristics, and so on. The purpose of the Road Network Database is to provide the data structures necessary to capture all of the information necessary about road networks so that a planner or control system on an autonomous vehicle can plan routes along the roadway at any level of abstraction. At one extreme, the database provides structures to represent information so that a low-level planner can develop detailed trajectories to navigate a vehicle over the span of a few meters. At the other extreme, the database provides structures to represent information so that a high-level planner can plan a course across a country. Each level of planning requires data at different levels of abstraction, and as such, the Road Network Database must accommodate these requirements.

Iconic Knowledge. Iconic knowledge is often spatial in nature and can be defined as two-dimensional or three-dimensional array data in which the dimensions of the array correspond to dimensions in physical space. The value of each element of the array may be Boolean data, a real number, or vector data representing a physical property such as light intensity, color, altitude, range, or density. Each element may also contain spatial or temporal gradients of intensity, color, range, or rate of motion. Each element may also contain a pointer to a geometric entity (such as an edge, vertex, surface, or object) to which the pixel belongs.

Examples of iconic knowledge used within 4D/RCS include digital terrain maps, sensor images, models of the kinematics of the machines being controlled, and knowledge of the spatial geometry of parts or other objects that are sensed and with which the machine interacts in some way. This is where objects and their relationship in space and time are modeled in such a way as to represent and preserve those spatial and temporal relationships, as in a map, image, or trajectory.

Within 4D/RCS, maps enhance the scope of the world model. Such iconic maps may take a variety of forms including survey and aerial

maps and may provide significant information about existing topology and structures. The higher levels in the 4D/RCS control hierarchy include feature and elevation data from a priori digital terrain maps such as information about roads, bridges, streams, woods, and buildings. This information needs to be registered and merged with data from the lower-level maps that are generated by sensors. Additionally, for incorporating a priori knowledge into the world model, some form of weighting is required, and this depends on how well the a priori data and the sensed information are registered. There is also the need to generate higher-resolution a priori terrain maps as the current survey maps are too coarse for autonomous driving. Another potential application for registering sensor data is the computation of ground truth.

Towards registering LADAR (laser range detection) range images to a priori maps, we have developed an iterative algorithm that can deal with false/spurious matches, occlusions, and outliers for UGV (unmanned ground vehicle) navigation (Madhavan and Messina 2003). The iterative registration algorithm can be summarized as follows: Given an initial motion transformation between two three-dimensional point sets, a set of correspondences are developed between data points in one set and the next. For each point in the first data set, we find the point in the second that is closest to it under the current transformation. It should be noted that correspondence between the two point sets is initially unknown and that point correspondences provided by sets of closest points is a reasonable approximation to the true point correspondence. From the set of correspondences, an incremental motion can be computed facilitating further alignment of the data points in one set to the other. This correspondence/compute motion process is iterated until a predetermined threshold termination condition.

A hybrid iterative algorithm has also been developed for registering three-dimensional LADAR range images obtained from unmanned aerial and ground vehicles (Madhavan, Hong, and Messina 2004). Combined with a feature-based approach, the algorithm was shown to produce accurate registration for the two sets of LADAR data. Registration of the UGV LADAR to the aerial survey map minimizes the dependency on GPS for position estimation, especially when the GPS estimates are unreliable or unavailable.

Results

Experimental validation of the 4D/RCS architecture and the knowledge representation within 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.

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 degrees wide and 20 degrees high with resolution of about 1/2 degree per pixel. It was mounted on a pan/tilt head that enabled it to look in the direction that the vehicle planned to drive. The LADAR detected the ground out to a range of about 20 m and detected 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 over 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 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 percent of both time and distance without any operator assistance. An extensive report of these experiments has been published (Camden et al. 2003), along with high-resolu-

tion ground truth data describing the terrain where the XUVs experienced difficulties (Witzgall, Cheok, and Gilsinn 2003).

Conclusion

We believe that 4D/RCS provides an excellent architecture in which to integrate multiple knowledge representation approaches to build cognitive models and intelligent systems that significantly advance the level of intelligence we can achieve. In this article, we have described how 4D/RCS supports multiple types of representations, ranging from iconic to symbolic and from declarative to procedural, and we have provided brief examples of how each of these representations is used in the context of autonomous driving. We have also shown not only how all of these knowledge representation formalisms fit into the node structure present at each level of the 4D/RCS hierarchy but also what role they play in the 4D/RCS methodologies.

It should be noted that the Demo III tests were performed in environments devoid of moving objects such as oncoming traffic, pedestrians, or other vehicles. In a dynamic environment, the autonomous vehicle would need to consider the actions, and possible future actions, of these types of objects in the environment. To address this, current and future efforts will focus on the development of predictive algorithms, leveraging the knowledge models found within this article, to predict the future actions of objects just as humans do when they drive. When humans drive, we often have expectations of how each object in the environment will move based upon the situation. For example, when a vehicle is approaching an object that is stopped in the road, we expect it to slow down and stop behind the object or try to pass it. When we see a vehicle with its blinker on, we expect it to turn or change lanes. When we see a vehicle traveling behind another vehicle at a constant speed, we expect it to continue traveling at that speed. The decisions that we make in our vehicle are largely a function of the assumptions we make about the behavior of other vehicles. It is believed that this level of "intelligence" is necessary to begin to achieve human-level AI.

In general, we believe that autonomous driving is an excellent topic for continued research on intelligent systems for the following reasons:

First, it is a problem domain for which there is a large potential user base, both in the military and civilian sectors. This translates into research funding.

Second, it is a problem domain where physical actuators and power systems are readily available. Wheeled and tracked vehicle technology is mature, inexpensive, and widely deployed.

Third, it is a problem domain for which the technology is ready. The invention of real-time LADAR imaging makes it possible to capture the three-dimensional geometry and dynamics of the world. This has broken the perception barrier. The continued exponential growth rate in computing power per dollar cost has brought the necessary computational power within the realm of economic viability. This has broken the cost barrier. Intelligent control theory has advanced to the point where the engineering of intelligent systems is feasible. This has broken the technology barrier.

Finally, it is a problem domain of fundamental scientific interest. Locomotion is perhaps the most basic of all behaviors in the biological world. Locomotion is essential to finding food and evading predators throughout the animal kingdom. The brains of all animate creatures have evolved under the pressures of natural selection in rewarding successful locomotion behavior. It is therefore, not unreasonable to suspect that building truly intelligent mobility systems will reveal fundamental new insights into the mysteries of how the mechanisms of brain give rise to the phenomena of intelligence, consciousness, and mind.

The 4D/RCS architecture and the autonomous driving domain provide an excellent opportunity to further advance research in human-level AI. Through the analysis of this domain, the process in which humans perceive and process information is becoming more evident. With the large amount of both moving and stationary objects in the environment and the inability of humans to be able to process every detail, challenges, such as focus of attention, are starting to be better understood and helping to drive research in promising directions. A better understanding of human driving is allowing methodologies, such as the state machine-based methodologies described in the state machine-based planning subsection, to be developed to better mimic the process that humans go through when they make decisions at different time horizons. These breakthroughs are also making evident the wealth of knowledge that humans use to make these decisions. The knowledge representation examples described in this article only begin to skim the surface of the knowledge representation techniques that are needed to achieve a level of humanlike intelligence. Continued research in this domain will help to

make these knowledge requirements more evident and help to drive future research with the ultimate goal of approaching human-level intelligence in AI systems.

Note

1. See the OWL Services Coalition, OWL-S 1.0 Release, 2003 (www.daml.org/services/owl-s/1.0/owl-s.pdf).

References

- Albus, J. 1971. A Theory of Cerebellar Function. *Mathematical Biosciences* 10(1–2): 25–61.
- Albus, J. 1997. The NIST Real-time Control System (RCS): An Approach to Intelligent Systems Research. *Journal of Experimental and Theoretical Artificial Intelligence* 9: 157–174.
- Albus, J. S. 1981. *Brain, Behavior, and Robotics*. New York: McGraw-Hill.
- Albus, J. S.; Balakirsky, S. B.; Bostelman, R. V.; Bunch, R.; Chang, T.; Dagalakis, N. G.; Falco, J. A.; Finkelshtein, R.; Gilsinn, J. D.; Hong, T. H.; Huang, H.-M.; Jacoff, A. S.; Juberts, M.; Kramer, T. R.; Lacaze, A. D.; Legowik, S. A.; Messina, E. R.; Michaloski, J. L.; Murphy, K. N.; Norcross, R. J.; Proctor, F. M.; Rippey, W. G.; Scott, H. A.; Shackelford, W. P.; Shneier, M. O.; Stouffer, K. A.; Szabo, S.; Wavering, A. J. 2002. *4D/RCS Version 2.0: A Reference Model Architecture for Unmanned Vehicle Systems*. NISTIR 6910. Gaithersburg, MD: National Institute of Standards and Technology, Office of Information Services.
- Albus, J. S.; and Meystel, A. 2001. *Engineering of Mind*. New York: John Wiley & Sons.
- Anderson, J. 1983. *The Architecture of Cognition*. Mahwah, NJ: Lawrence Erlbaum Associates.
- Archer, R.; Lebriere, C.; Warwick, W.; and Schunk, D. 2003. Integration of Task Network and Cognition Models to Support System Design. Paper presented at the Collaborative Technology Alliances Conference, April 29–May 1, College Park, MD.
- Archer, S.; and Adkins, R. 1999. *IMPRINT User's Guide*. Washington, D.C.: U.S. Army Research Laboratory, Human Research and Engineering Directorate.
- Balakirsky, S. 2003. *A Framework for Planning with Incrementally Created Graphs in Attributed Problem Spaces*. Berlin: IOS Press.
- Barbera, T.; Albus, J.; and Fitzgerald, M. 1979. Hierarchical Control of Robots Using Microcomputers. In *Proceedings of the 9th International Symposium on Industrial Robots*. Dearborn, MI: Society of Manufacturing Engineers.
- Barbera, T.; Fitzgerald, M.; Albus, J.; and Haynes, L. S. 1984. RCS: The NBS Real-Time Control System. Paper presented at the Robots Conference and Exposition, June 4–7, Detroit, Michigan.
- Camden, R.; Bodt, B.; Schipani, S.; Bornstein, J.; Phelps, R.; Runyon, T.; French, E.; and Shoemaker, C. 2003. Autonomous Mobility Technology Assessment Interim Report, Army Research Laboratory (ARL-MR-565). Washington, DC: Government Printing Office.
- Dickmanns, E. D. 1999. An Expectation-Based Multi-Focal Saccadic (EMS) Vision System for Vehicle Guidance. In *Proceedings of the 9th International Symposium on Robotics Research (ISRR'99)*. Stanford, CA: International Foundation of Robotics Research.
- Gonzalez, C. 2003. ACT-R Implementation of an Instance-Based Decision Making Theory. Paper presented at the Collaborative Technology Alliance Conference, April 29–May 1, College Park, MD.
- Kieras, D.; and Meyer, D. E. 1997. An Overview of the Epic Architecture for Cognition and Performance with Application to Human-Computer Interaction. *Human-Computer Interaction* 12(4): 391–438.
- Laird, J. E.; Newell, A.; and Rosenbloom, P. S. 1987. SOAR: An Architecture for General Intelligence. *Artificial Intelligence* 33(1): 1–64.
- Madhavan, R.; Hong, T.; and Messina, E. 2004. Temporal Range Registration for Unmanned Ground and Aerial Vehicles. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 3180–3187. Piscataway, NJ: Institute for Electrical and Electronics Engineers.
- Madhavan, R.; and Messina, E. 2003. Iterative Registration of 3D LADAR Data for Autonomous Navigation. In *Proceedings of the IEEE Intelligent Vehicles Symposium*, 186–191. Piscataway, NJ: Institute for Electrical and Electronics Engineers.
- Minton, S. N. 1990. Quantitative Results Concerning the Utility of Explanation-Based Learning. *Artificial Intelligence* 42(2–3): 363–391.
- Provine, R.; Uschold, M.; Smith, S.; Balakirsky, S.; and Schlenoff, C. 2004. Observations on the Use of Ontologies for Autonomous Vehicle Navigation Planning. In *Knowledge Representation and Ontologies for Autonomous Systems: Papers from the 2004 AAAI Spring Symposium*, 98–103. AAAI Technical Report SS-04-04, Menlo Park, CA, American Association for Artificial Intelligence. Revised and expanded edition to appear in the *Robotics and Autonomous Systems Journal*.
- Schlenoff, C.; Balakirsky, S.; Barbera, T.; Scrapper, C.; Ajot, J.; Hui, E.; and Paredes, M. 2004. The NIST Road Network Database: Version 1.0, National Institute of Standards and Technology (NIST) Internal Report 7136, Gaithersburg, MD.
- Schlenoff, C.; Washington, R.; and Barbera, T. 2004. Experiences in Developing an Intelligent Ground Vehicle (IGV) Ontology in Protégé. Paper presented at the 7th International Protégé Conference, Bethesda, MD, July 6–9.
- Shapiro, D.; and Langley, P. 1999. Controlling Physical Agents through Reactive Logic Programming. In *Proceedings of the Third International Conference on Autonomous Agents*, 386–387. New York: ACM Press.
- Witzgall, C.; Cheok, G.; and Gilsinn, D. 2003. Terrain Characterization from Ground-Based LADAR. Paper presented at the PerMIS 2003 Workshop, National Institute of Standards and Technology, September, Gaithersburg, MD.

Craig Schlenoff received his bachelor's degree in mechanical engineering from the University of Maryland, College Park and his master's degree in mechanical engineering from Rensselaer Polytechnic



Institute. He is a researcher in the Intelligent Systems Division at the National Institute of Standards and Technology. His research interests include knowledge representation, ontologies, and process specification, primarily applied to autonomous systems and manufacturing. He has recently served as

the program manager for the Process Engineering Program at NIST as well as the director of ontologies and domain knowledge at VerticalNet, Inc. He has served on the organizing and program committee of numerous knowledge representation and ontology related conferences and workshops, and has published more than 70 papers in related areas.



James Albus founded and led the Intelligent Systems Division at the National Institute of Standards and Technology for 20 years. He is currently a senior NIST Fellow pursuing research in autonomous vehicles, perception, knowledge representation, and intelligent behavior. He and Tony Barbera

were the original authors of the RCS (real-time control system). Albus was the primary author of the 4D/RCS reference model architecture developed for the Army Research Lab Experimental Unmanned Ground Vehicle program.



Elena Messina is a group leader in the Intelligent Systems Division at the National Institute of Standards and Technology. She manages numerous efforts to develop performance metrics and standards for robots and other intelligent systems. Her research interests also include knowledge representation,

software architectures, and tools for building robot control systems. Messina has a bachelor's degree in engineering science from the University of Cincinnati.



Anthony J. Barbera received his Ph.D. in biophysics from Johns Hopkins University in 1975. He has more than 30 years experience concentrated in the area of the development of a generic methodology (real-time control system [RCS]) for design and development of real-time, sensory-interactive,

knowledge-based control systems. Projects included a number of pilot control systems for factory automation (for the U.S. Postal Service, Boeing, and General Motors), and for military robotic vehicle systems for the U.S. Army (scout and logistics vehicles), U.S. Navy (UAVs and double-hull repair vehicles), and DARPA (submarine maneuvering control). Currently, he is involved in the task analysis of on-road driving and military tactical behaviors targeted to the design and implementation of autonomous vehicle control systems. He can be reached at barbera@nist.gov.

Proceedings of the Nineteenth International Florida Artificial Intelligence Research Society Conference

Edited by Geoff C. J. Sutcliffe

and Randy G. Goebel

906 pp., references, index, illus.,

\$125.00 softcover, ISBN 978-1-57735-261-7

To order, visit www.aaai.org/Press/Proceedings/flairs06.php



Raj Madhavan received the bachelor of engineering degree (electrical and electronics) from the College of Engineering, Anna University, India, and the master of engineering (research) degree from the Department of Engineering, the Australian National University, Australia. He received his

Ph.D. degree (field robotics) from the School of Aerospace, Mechanical, and Mechatronic Engineering, the University of Sydney, Australia. Since 2001, he has been a research associate with the Oak Ridge National Laboratory (ORNL) and is currently a guest researcher with the Intelligent Systems Division of the National Institute of Standards and Technology (NIST). His current research interests include autonomous vehicle navigation in complex unstructured environments, performance metrics for intelligent systems, distributed heterogeneous sensing, and systems and control theory. Madhavan has published more than 75 papers in refereed journals, book chapters, conference publications, and technical reports, and he can be reached at raj.madhavan@ieee.org.



Stephen Balakirsky is a researcher in the Knowledge Systems Group of the Intelligent Systems Division at the National Institute of Standards and Technology. He has more than 15 years of experience in multiple areas of robotic systems that include simulation, autonomous plan and behavior generation,

human-computer interfaces, automatic target acquisition, and image stabilization. Balakirsky's research interests include planning systems, simulation development environments, knowledge representations, world modeling, and architectures for autonomous systems.



The Conference on Innovative Applications of Artificial Intelligence

Now in Our 18th Year

**Please Help Us Celebrate
50 Years of AI at IAAI-06
Join Us in Boston!**

Details:

[www.aaai.org/Conferences/
IAAI/iaai06.php](http://www.aaai.org/Conferences/IAAI/iaai06.php)