Collaborative Tactical Behaviors for Autonomous Ground and Air Vehicles

James Albus, Anthony Barbera, Harry Scott, Stephen Balakirsky
National Institute of Standards and Technology
100 Bureau Drive, MS 8230
Gaithersburg, MD 20899

ABSTRACT

Tactical behaviors for autonomous ground and air vehicles are an area of high interest to the Army. They are critical for the inclusion of robots in the Future Combat System (FCS). Tactical behaviors can be defined at multiple levels: at the Company, Platoon, Section, and Vehicle echelons. They are currently being defined by the Army for the FCS Unit of Action. At all of these echelons, unmanned ground vehicles, unmanned air vehicles, and unattended ground sensors must collaborate with each other and with manned systems.

Research being conducted at the National Institute of Standards and Technology (NIST) and sponsored by the Army Research Lab is focused on defining the Four Dimensional Real-time Controls System (4D/RCS) reference model architecture for intelligent systems and developing a software engineering methodology for system design, integration, test and evaluation. This methodology generates detailed design requirements for perception, knowledge representation, decision making, and behavior generation processes that enable complex military tactics to be planned and executed by unmanned ground and air vehicles working in collaboration with manned systems.

Keywords: tactical behavior, 4D/RCS, task decomposition, autonomy, unmanned vehicles, intelligent systems

1. INTRODUCTION

The development of tactical behaviors within the Four Dimensional Real-time Controls System (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. The 4D/RCS framework consists of a hierarchy of intelligent control modules that can be readily configured to conform to a military style command and control structure. Each module consists of sensory processing, world modeling, value judgment, and behavior generation processes. Each module contains knowledge of two different kinds: 1) knowledge about entities, events, relationships, and situations contained in a knowledge database that is maintained by world modeling processes; and 2) knowledge of how to perform tasks embedded in behavior generation processes that include both rule-based and search-based procedures. Each module can be assigned specific duties and responsibilities. Each module can be assigned a specified span of control, both in terms of supervision of subordinates, and in terms of range and resolution in space and time. Typically, control loops (or OODA loops[1]) are closed through each 4D/RCS control module. Each echelon in the 4D/RCS hierarchy has a characteristic loop latency and update rate. Loop bandwidth tends to decrease at higher echelons in the hierarchy, while span of control increases at each higher echelon. More detail on the 4D/RCS architecture is available in a number of prior publications[1,2,3,4].

2. SCENARIO ANALYSIS

Initial 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

---

[*] OODA refers to Observe, Orient, Decide, and Act. The OODA loop was developed by Col John Boyd, USAF (Ret)

organization of the march column, and specifying the formation and movement technique. Within the tactical road march mission, attention has focused in 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. 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.

3. THE 4D/RCS METHODOLOGY

The 4D/RCS methodology focuses on task decomposition as the primary means of understanding the knowledge required for intelligent control. This approach can be summarized in six steps. These steps are shown in Figure 1, overlaid on a more detailed representation of the process to show the conceptual context for the steps. For a more detailed description of all the components of the approach shown in this figure, refer to Reference 5. The six summarized steps are:

1) 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.

2) 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. [http://www.isd.mel.gov/projects/rcs/, Gazi et al 2001]

3) 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 C2 (Command Control) hierarchy.

4) 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 world modeling process in each module. This knowledge database defines the set of situations that the sensory processing and world modeling processes must be able to detect and classify.

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

6) 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.

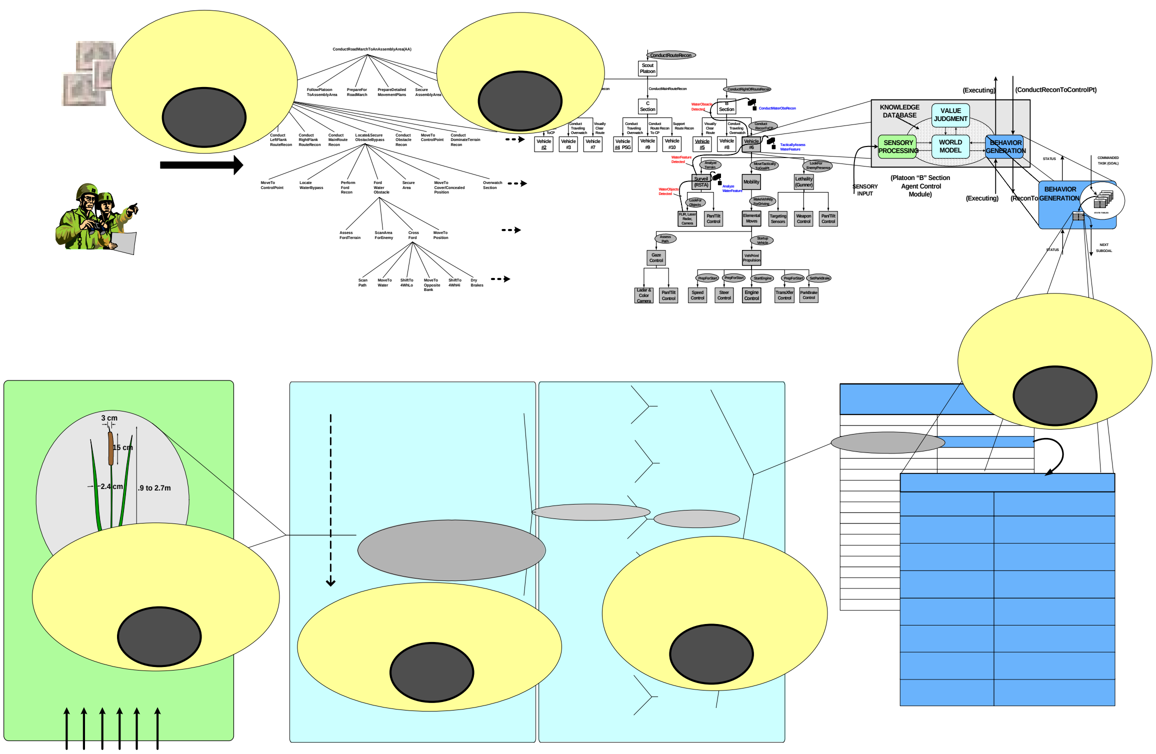

Figure 1. A summary of the 4D/RCS methodology for developing task knowledge.

The output of this 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).

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.

4. UGV/UAV/MANNED COORDINATION

The methodology described above can be used to develop requirements and guide software development for tactical behaviors for various combinations of manned and unmanned vehicles, including collaborative behavior between ground and air vehicles. It is currently being applied to the problem of collaboration between manned systems and unmanned ground and air vehicles prior to and during a route reconnaissance mission. Consider for example, a light cavalry platoon composed of three sections, each containing three manned High Mobility Multipurpose Wheeled Vehicles (HMMWVs), one unmanned ground vehicle (UGV), and one unmanned aerial vehicle (UAV). The activities of one section reconnoitering a stretch of road is illustrated in Figure 2.

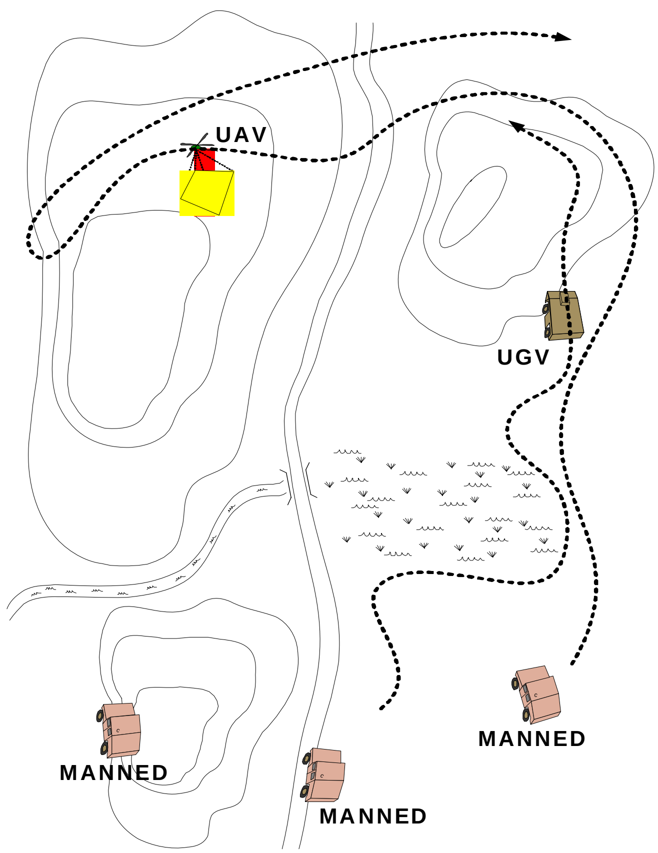

Figure 2. A diagram illustrating collaborative behavior of three manned and two unmanned elements reconnoitering a water obstacle and bridge during a route reconnaissance mission.

In this example, while the scout section is conducting a route reconnaissance, the HMMWV reconnoitering the right flank comes upon an unexpected water obstacle and the center HMMWV discovers a bridge. When this occurs, the two vehicle commanders report their findings to the section leader. The section leader then might command the manned vehicles to take up overwatch positions for near-side security. The section leader also commands the UAV to look for a route around the water obstacle, perhaps by sending hi-resolution color images data back to the section leader for manual viewing, and/or by scanning the ground with a LAser Detection And Ranging (LADAR) device to assess the topography. Once a potential by-pass to the marsh is located, the UAV is commanded to search the far side of the marsh and the region beyond the next terrain feature for evidence of enemy forces. The UGV might then be commanded to proceed through the bypass and establish an overwatch position on the far side of the next terrain feature. The UGV path can be automatically generated from the data returned from the UAV and approved by the section leader before being executed. Once the UGV is set in position, the UAV continues scanning for enemy activity further along the route while the manned elements perform manual reconnaissance of the marsh by-pass, and/or assess the load carrying capacity of the bridge.

5. SENSORY PROCESSING OF LADAR DATA

Among the capabilities that will be required for tactical behaviors in general and route reconnaissance in particular is the capability to drive on roads in traffic. On road driving is a technology that has been developed over the past 15 years. One of the earliest demonstrations of this technology was by Dickmanns and Graefe[6] Since then, Dickmanns and his students have demonstrated the ability to drive long distances at high speeds on German autobahn[7]. Jochem, Pomerleau, et al at Carnegie Mellon University demonstrated the ability to drive autonomously on freeways between Pittsburg, PA and San Diego, CA[8]. 98 % of the trip was executed under computer control. Commercial automobile manufacturers in Japan and Europe are developing intelligent cruise control systems that will enable drivers to put their vehicles on “autopilot” while on the freeway. LADAR sensors designed for freeway driving are under development for commercial vehicles that can detect and track moving and stationary vehicles, guard rails, utility poles, and pedestrians at distances up to 200 m.

However, freeway driving is a very limited subset of on-road driving. Entering and exiting freeways, negotiating intersections, recognizing and obeying traffic signs and signals, and coping with on-coming traffic, cross traffic, and pedestrians represent unsolved problems. Part of the research at NIST is focused on processing LADAR images that will enable safe and reliable autonomous driving on two lane roads with on-coming vehicles, cross traffic, and pedestrians.

Key to this effort is the use of the 4D/RCS architecture to bring high-level knowledge about road networks, geometry of the roadway, rules of the road, state of the self vehicle, and goals of the mission to bear on the processing of sensory information from advanced LADAR cameras. A simple example of this capability is illustrated in Figure 3.

In Figure 3a, a reflectance image from a high-resolution LADAR camera is shown. At the point in time when this image was taken, the knowledge database already contained the knowledge that the self vehicle was on the road, and the position of the road edges had previously been determined from data from color and LADAR cameras. Thus, the position of the road edges relative to the self vehicle could be projected onto the LADAR image as shown in Figure 3a. This enables the sensory processing algorithms to mask out all LADAR pixels that lie to the right of a vertical plane through the right road edge and to the left of a vertical plane through the left road edge. The result is shown in Figure 3b.

The next step is to compute the plane of the surface of the road. This can be achieved using the knowledge that the pixels immediately in front of the self vehicle lie on the road surface. By extending this knowledge to points further away in Figure 3b, knowledge of the road surface can be extended into the distance.

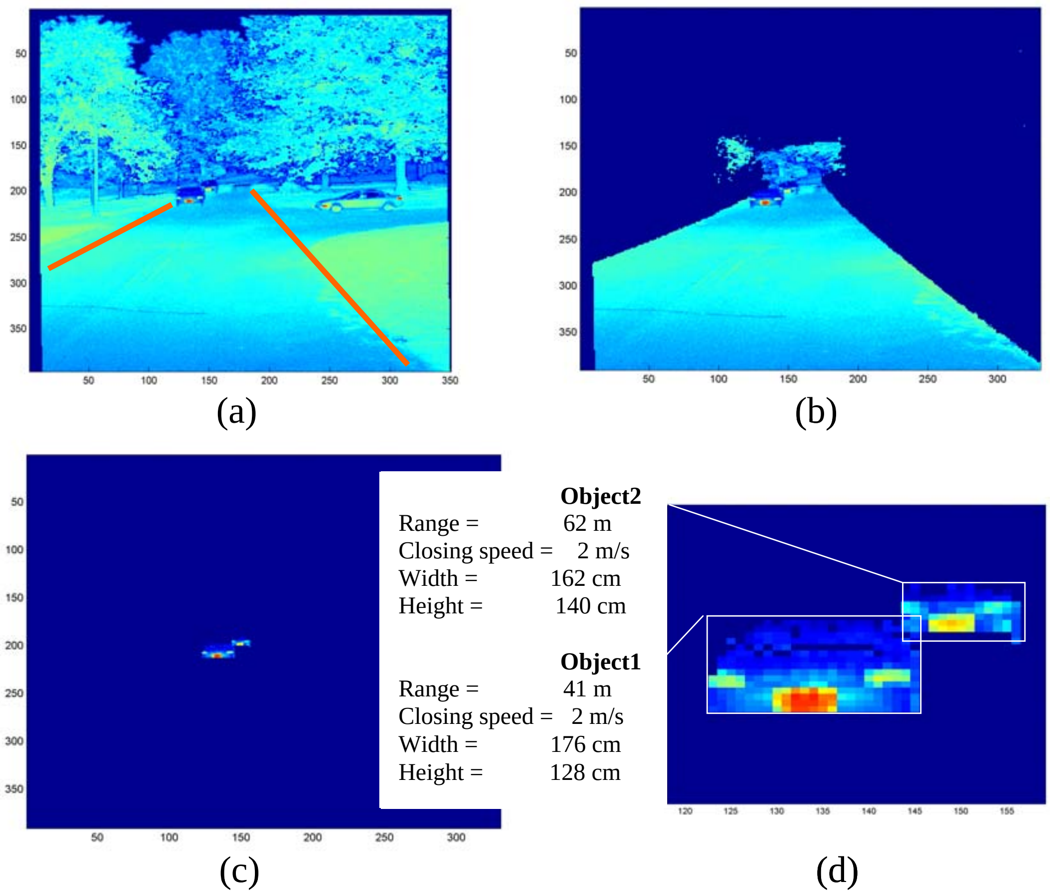

Figure 3. Processing of a LADAR image of on-coming traffic. The images are a false color representation of the LADAR reflectance image where red is bright, and blue is dark.

Next, it is possible to mask out all pixels except those that lie more than 20 cm, and less than 2 m, above the road surface. This results in the image shown in Figure 3c.

Finally, the range values of the remaining pixels can be clustered into two groups: one at about 41 m and a second at about 62 m. These groups correspond to pixels from the two cars on the road. It is then possible to compute attributes of each group. For example, x- and y-centroid and approximate width and height can be computed, and a bounding box placed around each group as shown in Figure 3d. The attributes of height and width can be used to classify the objects as cars, and distinguish them from trucks, motorcycles, or pedestrians.

When this process is repeated for two or more successive frames, it is possible to track the horizontal, vertical, and range velocities of the two objects relative to the self vehicle. Additional computations can determine that the two approaching cars are in the left lane and proceeding in a direction such that they will pass a safe distance to the left of the self vehicle. Calculation of relative velocity also enables the prediction of when each of the on-coming vehicles will pass the self vehicle.

6. THE MOAST FRAMEWORK: A REAL/VIRTUAL ENVIRONMENT

The development and test of tactical behaviors is a multi-disciplinary endeavor. It requires skills and expertise in fields as varied as sensor processing, knowledge representation, planning, execution and control, and even vehicle maintenance and repair. In addition, most behaviors require that there are multiple platforms that may require multiple safety personnel and large areas of real estate over which to operate. As a result, much of the development cycle is beyond the control of the individual algorithm developer. Vehicle up-time, safety personnel availability, and course availability play a large role in the development schedule. In addition, it may be difficult to isolate failures due to a lack of repeatable trials and the use of software modules that are being co-developed (which module has the bug?).

The need to develop on real hardware may severely limit the ability of resource-constrained institutions, or those without expertise in all the various areas, to fully participate in the field. In addition, research in a particular discipline may be limited by the current state-of-the-art in other, necessary disciplines. For example, when operating on a real system, the planning community can only construct plans based on features that the sensor processing community is able to detect. It is difficult or impossible to answer the question of how the system's plans would be affected if it could see feature 'x' at range 'y'. Because of this, planning researchers are unable to explore behaviors that require next generation sensor processing until that generation has arrived.

The Mobility Open Architecture Simulation and Tools (MOAST) framework has been developed as a real/virtual environment that allows researchers to concentrate their efforts in their particular area of expertise. This framework conforms to the NIST 4D/RCS architecture and allows simulated and real architectural components to function seamlessly in the same system. This permits not only the development of individual components, but also allows for component performance metrics to be developed and for the components to be evaluated under repeatable conditions. The framework is composed of high-fidelity and low-fidelity simulation systems, actual components under test, a detailed model of real-world terrain, a central knowledge repository, and architectural glue to tie all of the components together. MOAST also leverages a software development tool that facilitates the development of the overall RCS-based controller hierarchy, the creation of communication channels (NML) between the various components, and allows for real-time visualization and debugging of the functioning behavior.

MOAST has been designed to be a general-purpose framework that can be easily modified to become domain-specific. The value of such frameworks lies in their ability to reuse existing technologies and integrate their functionalities together into one complete set of tools. Specifically, MOAST's tools allow the framework to seamlessly integrate simulation subsystems with real robotic hardware subsystems. The goal is to allow the individual subsystems to each perform in the area where and when they do best. For example, simulation systems can replicate multiple platforms for the development of multi-platform behaviors. They allow for repeatable events, and may provide detailed system/event logging. In addition, by simulating the results of sensor processing, the potential benefits of detecting new features or utilizing novel sensing paradigms may be measured.

However, there is no substitute for real mobility, sensing, and communications. When available, real system components/subsystems must be able to plug into the MOAST framework and replace simulated subsystems. This is made possible through the architectural glue of the framework. This glue includes a reference model architecture that includes well-defined interfaces and communications protocols, and detailed specifications on individual subsystem input/output. A block diagram of an implemented system is shown in Figure 4. In the diagram, the light boxes are virtual components and the dark boxes are real systems. The system is composed of three vehicles; all of which have virtual sensing and low-level mobility. The figure only shows one of the vehicle's subsystems. The lowest level of the hierarchy (or echelon) shown in the diagram is subsystem and is composed of a mission and a mobility subsystem.

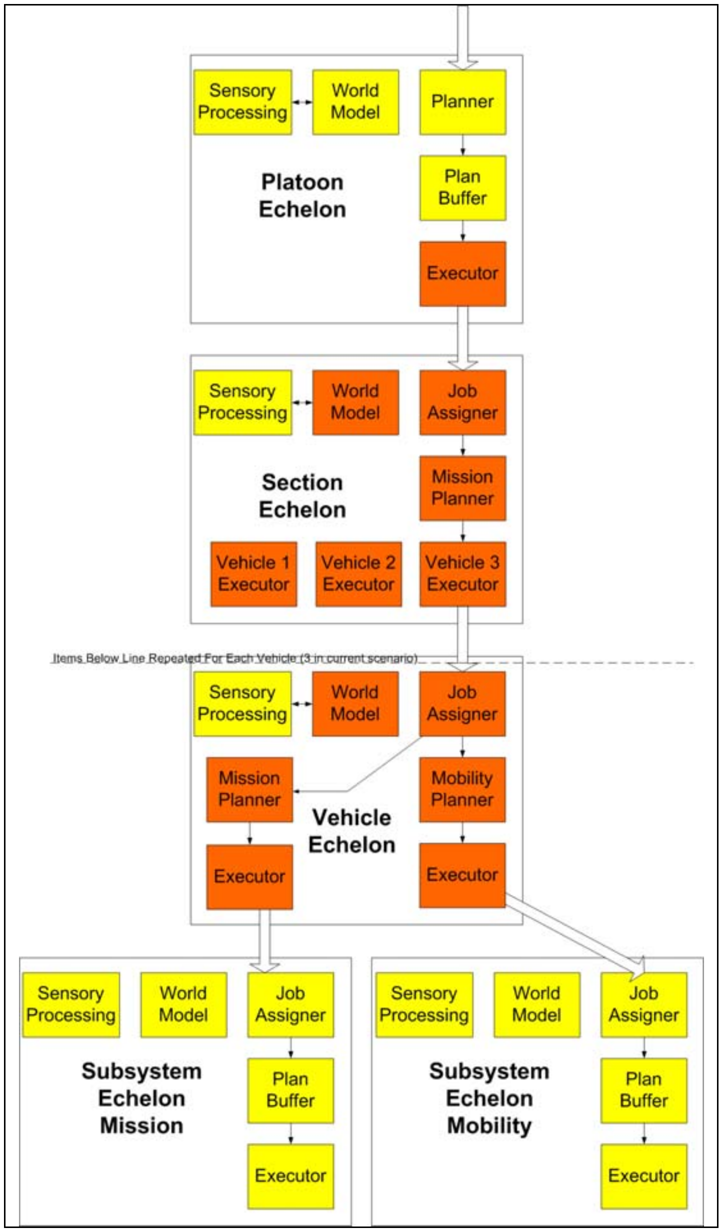

Figure 4: MOAST implementation including simulated (light) and real (dark) components.

6.1 System operation

In addition to providing support for the design and development of tactical behaviors, the MOAST framework provides an operating environment for the analysis and performance evaluation of the systems. Figure 5 depicts a three vehicle deployment around target "vehicle_I". The visualization is provided through One Semi-Automated Forces Testbed (OTBSAF)[9] that has been modified to include standard MOAST interfaces and communication channels. The tool that is inlaid in the bottom of the figure displays a real-time diagnostics tool that allows the user to view the functioning of all of the RCS modules in the system. At initialization time, this tool reads the communication header files and constructs a visualization of the complete RCS hierarchy and command and status interfaces. Component status is displayed (e.g. the last command executed, error status, component name), and any valid component status or command may be transmitted to the system. This allows for unit testing of an individual component or vehicle by providing command stimulus and status reports to the component. Figure 5 displays two robotic vehicles (shown as HMMWVs) and one manned vehicle (shown as a M1A1). However, to the functioning behavior and human observer, it is transparent as to which of these components are real systems and which are virtual entities.

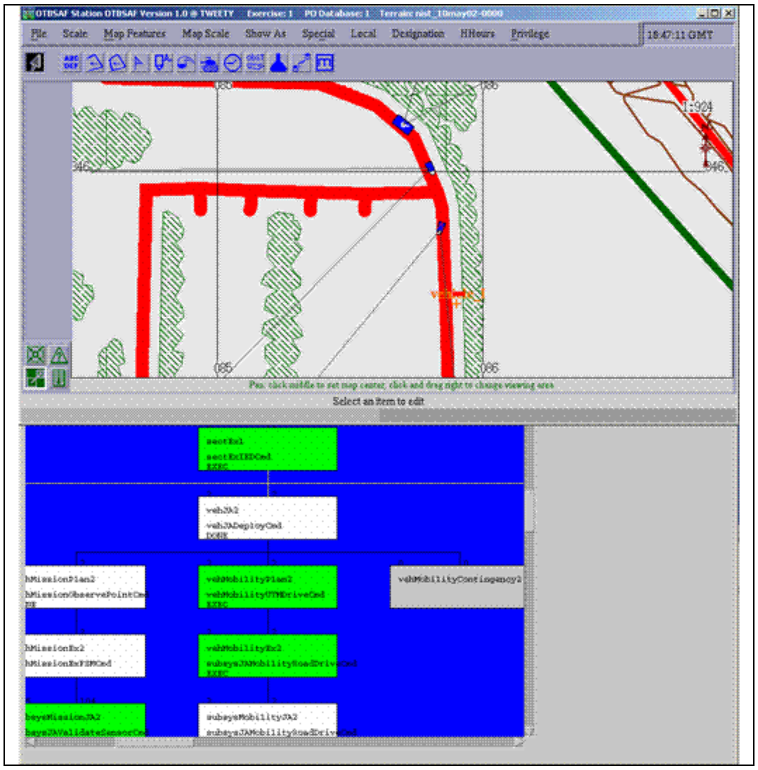

Figure 5: A 3 vehicle deployment as seen in the MOAST framework.

7. ARL/NIST HMMWV TESTBED

An autonomous HMMWV developed by NIST for the Army Research Laboratory (ARL) operating with the latest version of the 4D/RCS architecture is shown in Figure 6. The ARL/NIST HMMWV is equipped with a suite of LADAR and CCD cameras, INS and GPS systems, and internal sensors that provide real-time signals to a network of computers. This experimental vehicle is designed to provide an experimental platform for testing advanced sensors, and analyzing the performance of sensory processing, world modeling, planning and control algorithms for autonomous driving, both on-road and off-road. Data from this vehicle, coupled with information from a precise high-resolution digital representation of ground truth for the entire NIST campus and surrounding roads provides a capability for quantitative measurement of errors and latencies between internal representations and external realities.

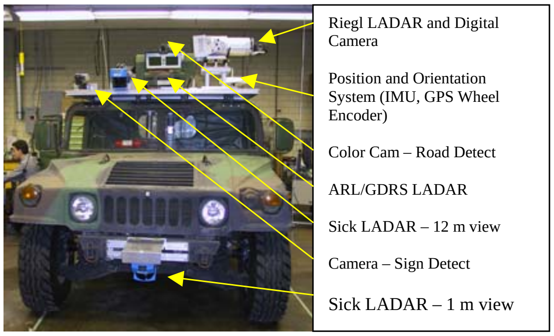

Figure 6. HMMWV Testbed Sensors

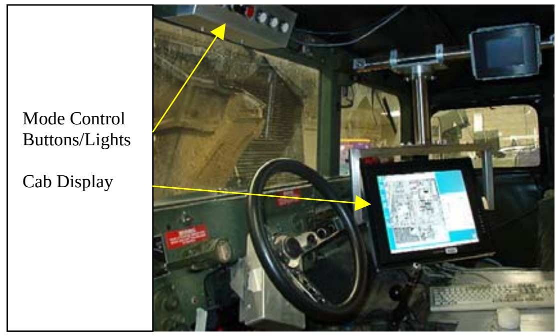

Figure 7. Mode Panel and Display

The ARL/NIST HMMWV testbed is configured to support integration and quantitative testing of a wide variety of components and behaviors. The major sensing components shown in Figure 6 include a Riegl[2] high-resolution LADAR with attached digital still camera, a LADAR capable of scan rates to support autonomous driving, two Sick line scan LADARs supporting curb detection experiments, a high-resolution color video camera in support of road detection, and another high-resolution color video camera in support of traffic sign detection. The various USB, Firewire, and ethernet data streams from these sensors have been routed through hubs in the vehicle cab to support easy access for experimentation. Pan and tilt capabilities are available on some of these sensors, with additional pan units planned for installation. Most of these sensors are installed on a shock-mounted platform above the vehicle cab.

Vehicle motion information, in support of both autonomous driving and performance measures, is provided by an Applanix position and orientation system at rates up to 200 Hz and with position precision available at up to 2 cm. High-precision position data are available in real-time when the HMMWV is used with a GPS differential base station (one has been installed on the NIST campus for this purpose) or via post processing. This system integrates an Inertial Measurement Unit (IMU) comprising three gyros and three accelerometers with two GPS receivers and a wheel encoder. Additional low-level sensors have been added or are planned for engine speed, throttle position, individual wheel speed, suspension motions and sensor motions.

To facilitate development and field tests, a suite of five PCs provide computing support for: 1) low-level mobility, supporting autonomous driving of constant curvature arcs through control of steer/brake/throttle/transmission/transfer case actuators and other vehicle functions, and status determination, 2) sensor processing, world modeling and higher level planning and control, 3) neural net processing for road detection, 4) 2D and 3D high performance graphics for visualization of world model representations, such as maps and data streams from sensors, and 5) general purpose computing, including hosting vendor-supplied user interface software (for Riegl, Applanix, etc.) and providing for large quantities of logged data. The PCs used for visualization and user interface tasks are intended to permit extensive test support without burdening the control computers. Communications among computers is via ethernet networks, with wireless ethernet available for off-vehicle communication. NML provides the messaging channels that move data between computational modules.

Ease of use, efficiency and safety for developers are enhanced by a startup and mode selection procedure in which a series of simple buttons and lights leads the operator through emergency stop, manual, and computer controlled modes of the testbed. Operator input is via the button/light panel above the windshield, shown in Figure 7. Status is provided by lighted buttons, and by messages on the large, daylight-readable display in the cab. All computers on the vehicle, as well as additional laptops, may have their displays routed to this display panel.
Electrical power is provided by an on-board 5500 w diesel generator. Power distribution is flexible, with eight individually breaker-protected circuits. Source power is also flexible, with potential sources including one or two generators, and 110 VAC or 220 VAC shore power.

---

[2] The identification of certain commercial products or companies does not imply recommendations or endorsement by NIST.

ACKNOWLEDGEMENTS

This work was supported in part by the Army Reserarch Lab's program in unmanned vehicles (PM. C. Shoemaker).

REFERENCES

1. Albus, J. 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. Albus, J. and Meystel, A., 2001, *Engineering of Mind: An Introduction to the Science of Intelligent Systems*. New York, NY: John Wiley & Sons, Inc.

3. Albus, J.S., "4-D/RCS: A Reference Model Architecture for Demo III," *Proceedings of the SPIE Vol. 3693 AeroSense Session on Unmanned Ground Vehicle Technology,* Orlando, FL, April 7-8, 1999.

4. Gazi, V., Moore, M.L., Passino, K.M., Shackleford, W.P., Proctor, F.M., and Albus, J.S., *The RCS Handbook: Tools for Real Time Control Systems Software Development,* John Wiley & Sons, NY, 2001.

5. Barbera, A., Albus, J., Messina, E., Schlenoff, C., and Horst, J., "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.

6. E. D. Dickmanns, and V. Graefe, a) Dynamic monocular machine vision, b) Application of dynamic monocular machine vision. Jounal of Machine Vision & Applications, Springer-Int., pp223-261, Nov, 1988.

7. Dickmanns, E., "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, 199*9.*

8. Jochem, Todd M., Pomerleau, Dean A., and Thorpe, Charles E., "Vision Guided Lane Transition," *IEEE Symposium on Intelligent Vehicles, September 25-26, Detroit, Michigan, USA, 1995.*

9. Parsons, D., and Whittman Jr., R., "OneSAF Tools and Processes Promoting and Supporting a Distributed Development Environment for Multi-Domain Modeling and Simulation Community," *Proceedings of the Spring 2004 SIW Conference*, 2004.