Proceedings of the SPIE AreoSense 98 Conference, Orlando, FL, April 13 – 17, 1998.

Advanced MMI/MP for Demo III XUVs

Stephen B. Balakirsky[a], Michael J. Salonish[b], Swati D. Allen[b], Elena Messina[a], Jose Salinas[b]
[a]National Institute of Standards and Technology*, Gaithersburg, MD 20899
[b]Army Research Laboratory, Adelphi, MD 20873

ABSTRACT

This paper outlines the goals and work accomplished thus far for both the man-machine interface (MMI) and mission planning (MP) elements of the eXperimental Unmanned Vehicle (XUV) program. It is the goal of the XUV program to make available to the user an interface and tools that will allow for seamless transition between mission planning, rehearsal, and execution on multiple collaborating autonomous vehicles in a platoon group.

For scalability purposes, the final system will have two different levels of operator control unit (OCU). The Tier 1 controller will be a web-based control station, preferably a laptop, that will allow the user to control and view all major functions of the robotic vehicle. Tier 1 control will be accomplished via a web client interface executing Java applications. By using an independent web client for each vehicle, any number of vehicles may be controlled. The Tier 2 station will encompass the Tier 1 functionality with the additional capability of complex mission planning and scene visualization. The Tier 2 OCU will integrate control of the platoon (up to four vehicles) into a tightly coupled system executing on a two-dimensional and three-dimensional rendering and control engine. In addition, an inter-vehicle interface is defined for collaboration at the vehicle level.

Initial implementations of both the Tier 1 and Tier 2 OCUs exist and are in use today. The Tier 1 OCU is written in Java and provides vehicle level control of the XUVs. The Tier 2 system is based on the Combat Information Processor (CIP) and Virtual Geographic Information System (VGIS) from the Army Research Laboratory (ARL). The Tier 2 system is capable of both 2D and 3D visualization of the battlefield environment and provides vehicle path planning for the XUVs.

The Four Dimensional Real-Time Control System (4-D/RCS) reference model architecture is used as the basis for the division of decision and mission planning responsibilities among the control stations and the robotic vehicles.

Keywords: Robotics Demo III, Mission Planning, Vehicle Control, Robotic Vehicles

INTRODUCTION

This paper outlines the goals and work accomplished thus far for both the man-machine interface (MMI) and mission planning (MP) elements of the eXperimental Unmanned Vehicle (XUV) program. The XUV program is the government's Concerted Technology Thrust (CTT) for the Office of the Secretary of Defense's Robotics Demo III. Through this program, government laboratories are striving to develop the technology necessary for a group of unmanned ground vehicles (UGVs) to perform the mission of a scout platoon. This mission includes such activities as autonomous high-speed on/off road driving in military formations, area and route reconnaissance, and calling for fire on enemy locations. In addition to these tasks, a commander must be able to communicate with the robotic platoon in a similar manner as he would communicate with his existing troops. This requires an easy-to-use, intuitive MMI and an embedded mission planning capability. The mission planning system must be able to accept a military platoon-level plan (a plan for a group of up to four vehicles) and decompose that plan into coordinated individual plans for each vehicle.

A conscious decision has been made by the XUV community to combine the areas of MMI and MP into a single entity. The reason behind this decision is to maintain the close relationship that must exist between mission planning and execution. It is the goal of the MMI/MP XUV community to make available to the user an interface and tools that allow for seamless

---

No approval or endorsement of any commercial product by the National Institute of Standards and Technology is intended or implied. Certain commercial equipment, instruments, or materials are identified in this report in order to facilitate understanding. Such identification does not imply recommendation or endorsement by the National Institute of Standards and Technology, nor does it imply that the materials or equipment identified are necessarily the best available for the purpose.

transition between mission planning, rehearsal, and execution on multiple collaborating autonomous vehicles in a platoon group. This platoon group may function in either the real or virtual (simulation) world. This goal will be achieved through the insertion of technology currently being developed under a variety of government programs along with innovative research at U.S. Government Laboratories.

ARCHITECTURE

The current design for the Demo III XUV system allows for varying degrees of operator intervention in the control of the XUV platforms and is consistent with the Demo III Reference Model Architecture[1] (4-D/RCS). This architecture defines functional processes at each level of a hierarchy that correspond to the typical operational units in a military organization. This enables the 4-D/RCS architecture to map directly into the military command and control structure to which the Demo III vehicles will be assigned. The result of this mapping is a vehicle system that is understandable and intuitive for the human commander and integrates easily into battle space visualization and simulation systems. Through the use of the OCU, the operator may interface with the vehicle systems at several different levels of the control hierarchy. This allows the user to take command of an individual vehicle and plan a specific route (vehicle level control), or to give a general military order to a group of vehicles (section or platoon-level control). In the first case, the user will interact with the vehicle as a vehicle commander. In this capacity, he will command the vehicle to drive on a certain route, and observe specific areas. The user will also maintain constant over-watch of vehicle location and status. In the second case, the user will interact with the vehicle as a platoon commander. In this capacity, he will specify military control measures and an objective for the vehicles to accomplish. OCU and vehicle-based planning software will then plan vehicle routes and objectives and only notify the user when specific mission objectives have been reached. This architecture breaks the control station into two independent conceptual units.

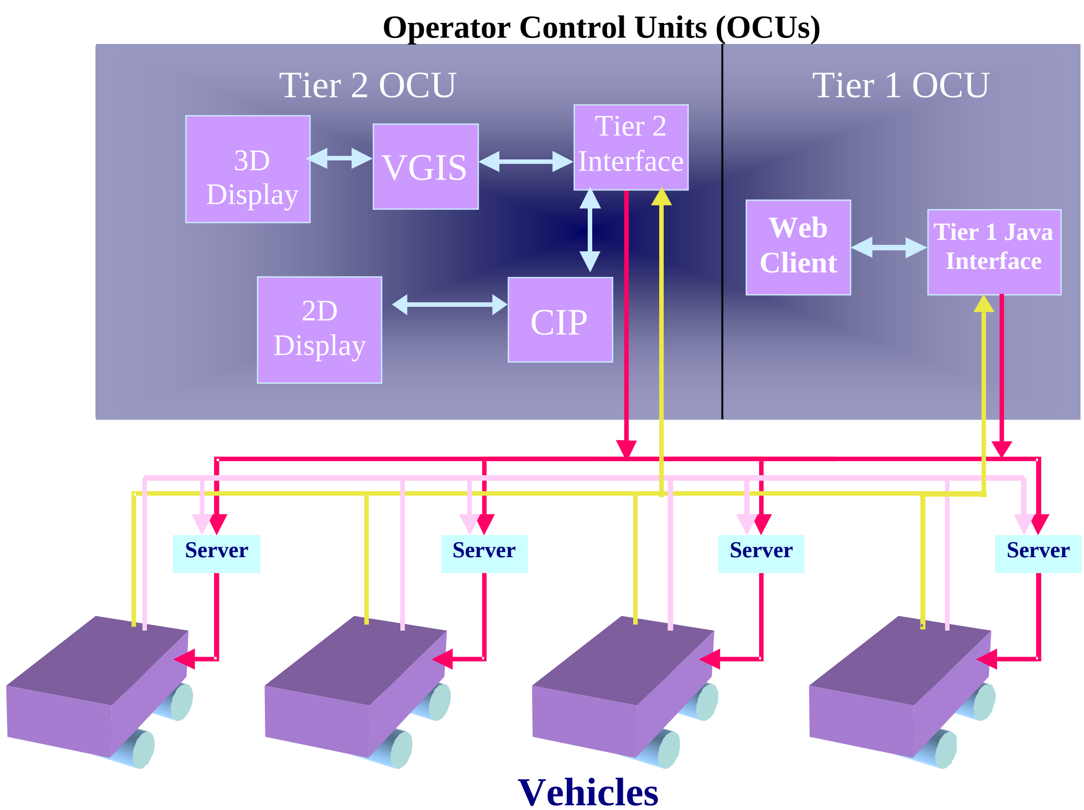

Figure 1: System configuration and functional decomposition

In addition to being conceptually independent, the OCU hardware is divided into two independent hardware stations as shown in Figure 1. Although the two systems are physically independent, they are logically integrated into the reference architecture via a communications server to implement the 4-D/RCS control hierarchy between the vehicle and the control station. This control structure allows the OCU to set global mission objectives and priorities, while not interfering with the local behavior of the vehicle(s). Although the OCU controls the global behavior, the vehicle(s) are capable of autonomy (within user set limits) when access to the OCU is prohibited either doctrinally (i.e., communications shutdown) or physically (i.e., bandwidth overload or no line of sight).

4-D/RCS provides a reference model architecture for the design, engineering, integration, and test of intelligent, supervised-autonomy controllers. As a reference model architecture, 4-D/RCS defines the functional elements that are used to construct intelligent vehicle systems. The functional elements, or building blocks, are assembled within computational nodes in a hierarchical fashion based on the specifics of the system to be built. As shown in the example vehicle hierarchy in Figure 2, each level has its own temporal scope. The basic functional elements comprising a computing node are behavior generation (BG), sensory processing (SP), world modeling (WM), and value judgment (VJ). These functional elements prescribe basic capabilities that must exist in order to achieve intelligent control. The capabilities exist at each level of the hierarchy, albeit performing different algorithms or dealing with different data or resolutions appropriate to their scope. The functional elements in a 4-D/RCS compute node are briefly discussed below.

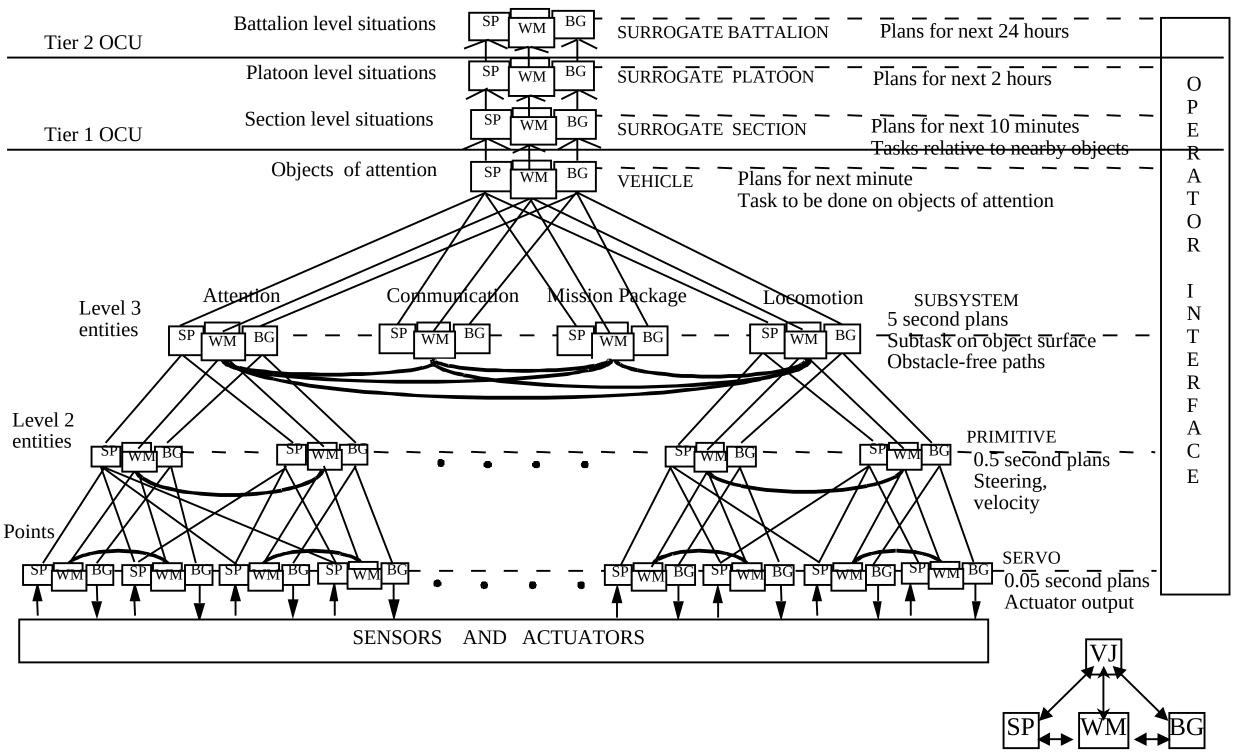

Figure 2: 4-D/RCS hierarchy.

A 4-D/RCS node

Behavior generation accepts task commands with goals and priorities, formulates and/or selects plans, and controls action. The behavior generation module is responsible for developing a plan based on inputs from the world model and the user. Each behavior generation process accepts tasks from its superior level and decomposes these tasks into subtasks for its subordinate level(s). The coordination and tracking of these sub-tasks among the multiple subordinates is also the responsibility of a behavior generation module. The behavior generation module receives periodic status reports from the subordinates and makes corrections to their plans to ensure that the goals are accomplished.

The sensory processing module at each level of the hierarchy produces the results of signal and image processing which has been performed on sensory processing outputs from lower levels of the hierarchy. These outputs are stored in the world model databases for use in behavior generation and decision-making. Sensors include vision and laser range imagers, inertial navigation, and vehicle status indicators such as fuel level.

The world modeling process constructs, maintains, and uses world model knowledge to support sensory processing and behavior generation. The world model at each level of the hierarchy contains information about the world that is obtained from sensor processing or external communications. It also encompasses other longer-term forms of information that are contained in the knowledge database, such as terrain maps, models of enemy vehicles, or military tactics. This information is of the appropriate granularity to allow cost maps to be generated that will allow for intelligent planning decisions up to the planning horizon for the particular level.

The value judgment functional element computes cost, risk, and benefit of actions and plans, estimates the importance and value of objects, events, and situations, assesses the reliability of information, and calculates the rewarding or punishing effects of perceived states and events. Value judgment evaluates perceived and planned situations using simulations thereby enabling behavior generation to select goals and set priorities.

OPERATOR CONTROL UNITS (OCUS)

For most past robotic programs, each class of vehicle required its own specialized control station. This station usually incorporated proprietary software and hardware, and was only capable of controlling a single class of robotic vehicle. This architecture for constructing control stations prohibits the timely incorporation of third party hardware and software, and prevents the workstation from scaling to the needs of different users.

In the XUV program, the control station hardware is a modular open system design, based on commercial platforms (such as Sun stations and laptop 486 based computers). The control station software is based on the 4-D/RCS architecture implemented through a client-server model. The use of the client-server model allows for identical communication paths for both control stations to vehicle and vehicle to vehicle communication. This communication, both intra- and inter-vehicle is accommodated over standard sockets using commercial network protocols. All of the above allows for the integrating of third party software/hardware and redistribution of existing tasks with little or no modification of the existing system.

For scalability purposes, the system has two different levels of operator control station. The Tier 1 controller is a web-based control station running on a laptop computer that allows the user to control and view all major functions of the robotic vehicle at the vehicle level and below. The Tier 2 station, running on a high-end graphics workstation, encompasses the Tier 1 functionality with the additional capability of complex mission planning (section and platoon-levels of 4-D/RCS, mission rehearsal) and mission visualization (situation display, scene visualization, and mission monitoring).

Each OCU will be capable of controlling a set of up to four autonomous vehicles (as dictated by Demo III requirements). The Tier 1 control is accomplished via an internet browser interface executing a Java applet. By using an independent browser for each vehicle, any number of vehicles may be controlled. The Tier 2 OCU integrates control of up to four vehicles into a tightly coupled system executing on a two-dimensional and three-dimensional rendering and control engine. When work on the Tier 2 OCU is complete, the commander will be able to enter his standard control measures and fragmentary orders which the station will automatically interpret for robotic use.

TIER 1 CONTROL STATION

The Tier 1 control station provides an interface for vehicle mission planning and data collection. It is used to perform system checks, and to command vehicle level missions. Through the use of this station, the user will be able to view planning information such as the planned routes and surveillance areas of the vehicle. The user will also be able to view real-time status from the vehicle that includes the actual route taken and vehicle health status. Displays will also show vehicles and obstacles that have been encountered during the mission.

The current version of software release for the Tier 1 controller is capable of limited mission/vehicle status display, and limited mission specification. To make the code as portable as possible, it was decided that the interface would be web based, and implemented in Java. The current vehicle status display may be seen in Figure 3. This display may be broken down into three independent areas; the vehicle orientation indicator, vehicle status box, and map display area. The vehicle orientation indicator displays a three-dimensional pictograph of the vehicle orientation. As the vehicle traverses terrain, this pictograph follows the actual vehicle motion. The vehicle status box displays text information from critical vehicle subsystems. This information may be automatically monitored to alert the user of abnormal values. The map display area displays the route that the vehicle has traversed. Future enhancements will allow the map area to also display obstacle information as well as planned route and map images. All of the screen areas are independent Java applets with refresh times that may be specified by the user. This allows for a mix of continuous and intermittent data update to best balance channel bandwidth verses status reporting requirements. In addition, the independent applets allow for a toolbox approach to the Tier 1 interface. The interface may be quickly and easily customized for each individual user.

By incorporating a web server as the basic gateway into the vehicle, we allow any computer system that is capable of running a web browser to obtain data. If this paradigm is followed, then no unique control station software needs to be developed. The vehicle simply downloads its control interfaces to the user.

TIER 2 CONTROL STATION

The Tier 2 OCU will encompass the Tier 1 functionality with the additional capabilities of complex mission planning, mission rehearsal, situation display, scene visualization, and mission monitoring. In addition to hosting the primary platoon planning system, the Tier 2 OCU provides the interface between the vehicles and the command and control architecture in

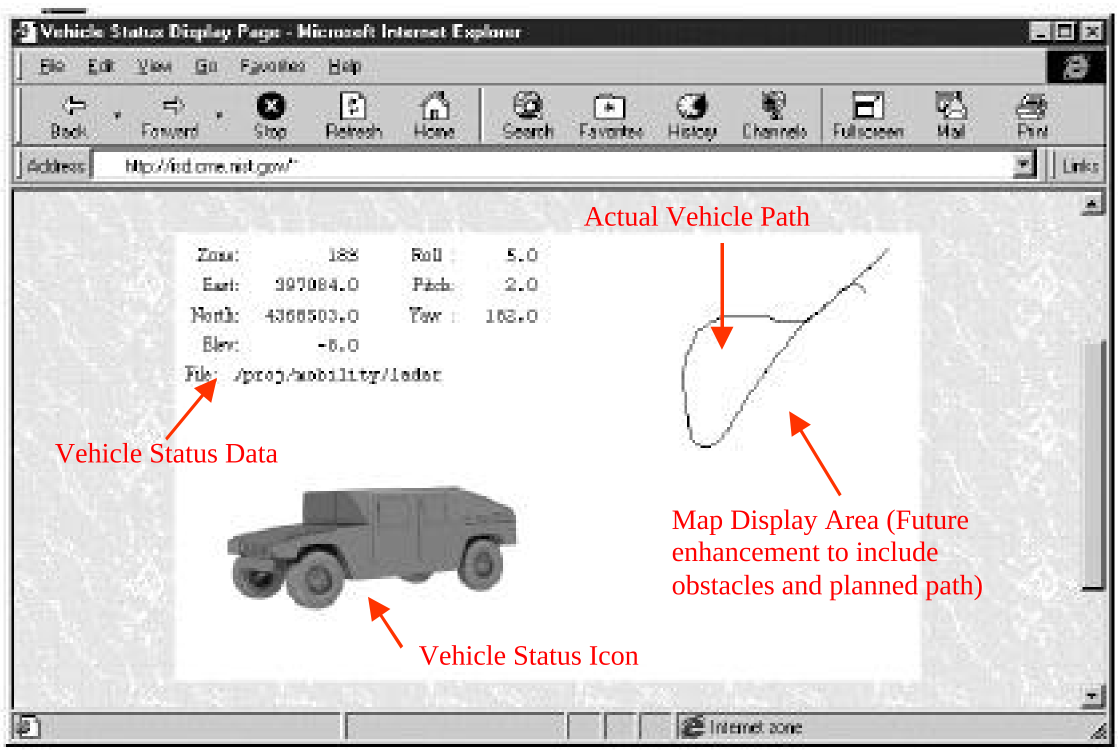

Figure 3: Tier 1 status display.

which the vehicles are being utilized. Each Tier 2 station (multiple Tier 2 stations may be deployed but are not required) will have the capacity for controlling a set of up to four autonomous vehicles that will be used for planning complex platoon-level missions consistent with the 4-D/RCS architecture.

The mission planning application, along with the wireless communications capabilities, will allow the Tier 2 station to develop, monitor, evaluate and re-plan missions for the autonomous vehicles involved in the activity. The application will assign a formation and suggested vehicle-level tasks while allowing the vehicles to arbitrate for position and tasking amongst themselves during operation. As the missions are executed, the high-resolution imagery/terrain data that may be collected will be incorporated into the terrain databases for display on the map screens, 2D and 3D, and for use in future mission planning.

The Tier 2 OCU will leverage capabilities that exist in the systems developed and deployed by the Army Research Laboratory. The rendering engines for the 2D and 3D displays, as well as the communications network and digital terrain server, will be based on existing packages. These packages include the Combat Information Processor (CIP) and associated utilities and servers, and the Virtual Geographic Information System (VGIS).

BACKGROUND ON COMBAT INFORMATION PROCESSOR (CIP)

The CIP was developed as an architecture to perform the integration of technologies and applications that perform tasks of military significance. The CIP includes a set of servers, a front end 2D mapping application, and a set of applications that access those servers to perform a wide variety of tasks. These servers map easily into the 4-D/RCS architecture as shown in Figure 4. A few of these tasks include horizontal battlefield integration, multi-sensor fusion, real-time moving-target indicator (MTI) display, and situation display/development. The servers are used to cache data from existing internal and external databases such that the applications/clients have a common interface to access data regardless of the source of the information. In addition to supporting traditional query capabilities, the servers also have the capability to push data to clients that have previously registered for data. The servers support multiple clients in a distributed environment and the data categories can logically be viewed as terrain, environmental, and tactical.

The terrain server can be further decomposed into an elevation server and a feature server. The elevation server provides an interface to the standard National Imagery and Mapping Agency (NIMA) Digital Terrain Elevation Data (DTED) product. The feature server provides an interface to an in-house representation of vector product formatted thematic data (i.e., roads, lakes, brush, etc.). There are two associated libraries, one that calculates trafficability based on terrain and entity type, and one that stores features in quadtree format for efficient feature data retrieval. The terrain server and its associated libraries are utilized by the CIP applications to augment reasoning algorithms with real-world data and to provide the user with more meaningful value-added decision-making aides for the battlefield.

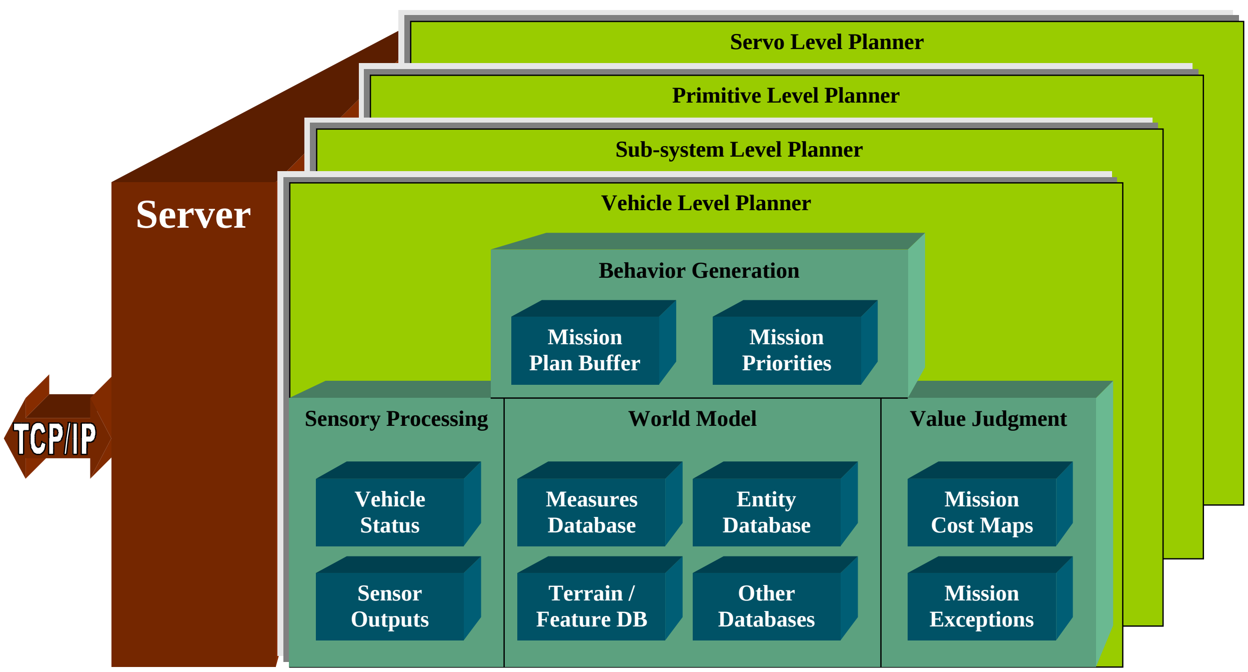

Figure 4: Vehicle software architecture.

Two environmental servers comprise the weather server and are implemented for weather and Nuclear/Biological/Chemical (NBC) forecasting and modeling. The weather server provides an interface for retrieving both coarse (meso)-scale weather patterns and forecasts, in addition to vector and scalar weather patterns over high-resolution terrain. Generated data include directional wind-vector data over a set of predefined atmospheric layers and a set of scalar values representing temperature and humidity over the same layers with additional values for ground elevations. A modeling module in the server allows for fine scale weather forecasting using a given mesoscale input forecast over a given tactical area of interest. Weather data is stored using an open architecture based on the Network Common Data Form (NetCDF) format which has been implemented on several different hardware architectures for increased portability and maintainability. For generating NBC transport and diffusion forecasts, the weather server high-resolution forecasts are mapped into an NBC server. It contains NBC parameters and agent/munition combination tables used for model generation over the area of interest. Output NBC isosurface contours

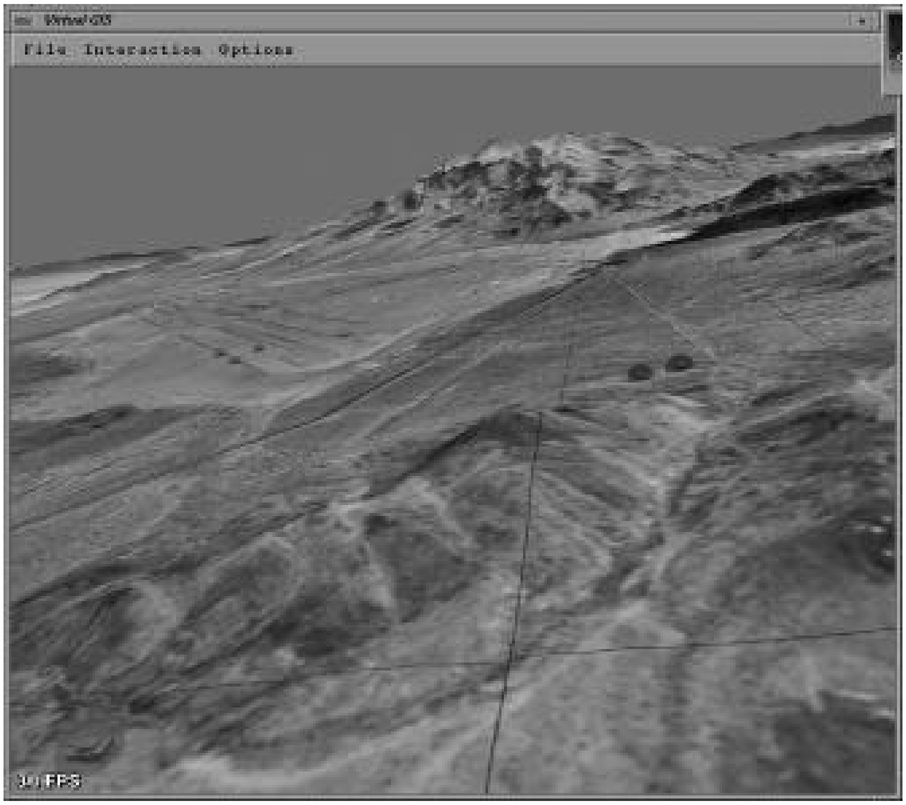

Figure 5: VGIS display output.

and ground values are sent to the client for rendering on the given display engine. Values computed include dosage levels, concentrations, and lethality percentages over the affected area of interest.

The tactical server provides a mechanism to interface with external databases allowing the CIP to provide a global situation display. The tactical server consists of an entity server and a control measures server. The entity server provides data on tactical entities currently active in the battlefield. The control measures server maintains the tactical control measures data that indicate past, current, and planned tactical positions or events.

The front-end 2D mapping application uses the Defense Information Infrastructure (DII) Joint Mapping Toolkit (JMTK) as the map display manager. All display-related applications are launched from the main window and utilize the JMTK Application Programming Interfaces (APIs) to interface with the display. The existing functional applications that are available on the CIP handle map management, battle planning, and simulation control. The Graphical User Interface (GUI) of the CIP consists of a standard Motif/X11 menuing system.

BACKGROUND ON VGIS[2]

The Virtual Geographical Information System (VGIS) was developed as an approach to display terrain information in a three dimensional environment. The system provides a program that is used for visualizing terrain models that contain different data sets that represent different aspects, resolution, and imagery values for given terrain areas. A typical output of the VGIS system may be seen in Figure 5.

The system is composed of a core program that uses a quadtree database representation of the terrain models to efficiently navigate the world in 3D space. Because of the 3D nature of VGIS, an immersive experience for the user is achieved when executing applications on the given terrain.

The advantage of this system is the ability to navigate in real time through a rendered world that is composed of the different terrain models within the database. The quadtree implementation of the database allows for lower resolution data to be stored at the higher nodes within the tree. As the user approaches the terrain, higher resolution nodes at the lower levels of the tree will be rendered by the system until the highest resolution possible from the terrain model is displayed. New data can be added to the database in real-time without interrupting operation to allow for additions of terrain data from sensors in the field.

DESIGN OF THE PLANNING AND EXECUTION SYSTEM

As shown in Figure 2, the 4-D/RCS hierarchy for a vehicle has several levels, with servos at the lowest level, and surrogate companies or higher potentially at the highest level. A surrogate chain of command resides on the vehicle, and is normally dormant, but is equipped to perform higher-level functions if required. Surrogate levels may be activated if a vehicle is designated as the lead in a formation or other collaborative situations involving multiple vehicles. The functions of the platoon-level planning and control can occur on either the OCU or on a vehicle designated as the platoon leader. In the following discussion, for simplicity, we will focus on two tiers of this military hierarchy that will be implemented for Demo III: platoon and vehicle.

Each level of the hierarchy performs planning and control over a different spatial and temporal range, appropriate to its level of command and control. This enables the implementation of systems having great complexity, yet composed of individual nodes of controlled complexity. At no level does a node have to cope with both broad scope and high level of detail. The platoon commander creates plans for about an hour into the future, re-planning every five minutes with subordinate feedback to keep the accomplishment of the desired goals on track. Commands passed to subordinates (in our discussion, the vehicles) are at about a five-minute resolution. Therefore, the distance between waypoints varies, depending on how fast the vehicles are supposed to move. The level of detail that a platoon-level is concerned with is coarser than that of the vehicle level, however, the platoon-level's world model covers a larger area. Platoons typically work with maps of 30 meter resolution that cover about 100 km allowing for a more global view and accommodating the higher level mission planning.

The platoon-level behavior generation module receives its orders from its superior, which could be another level of software, or a human. The platoon-level planner breaks its mission into higher-granularity military behaviors, which provide a course of action for the platoon of vehicles. The orders are decomposed into job assignments for each vehicle and a schedule of activities is produced for the vehicles. Sensory processing integrates information received from its subordinate levels, which may include location and status of enemy and friendly forces. Major terrain features, military tactics, other vehicles, and current military intelligence are all factored into the building of the plans to accomplish the mission. Value judgment evaluates tactical options for achieving objectives. For simpler tasks, such as traveling to a location, the platoon-level does not have to plan extensively for the cooperative behavior among vehicles. A formation appropriate to the situation may be chosen given military doctrine and individual vehicle plans are built around this formation. Vehicles are given the responsibility of maintaining the formation.

For more complex military missions, such as area reconnaissance, the platoon-level must split up the area among the vehicles, based on mission timing requirements and sensor capabilities on the vehicles, to ensure maximal coverage of the area. A set of waypoints designed to cover each vehicle's area of responsibility is downloaded to the individual vehicles. Databases, such as terrain maps, to aid the vehicles in accomplishing their mission may also be downloaded. As the mission unfolds, vehicles may regularly broadcast status to the platoon-level commander and receive modifications to their plans if needed. If one of the vehicles is disabled, re-plannning is performed by the platoon-level to have the other vehicles cover its area. Deviations from the given plan always have some measure of tolerance included, defining an allowable error band. A vehicle is therefore allowed to stray from its designated path or schedule by a certain amount. If it strays beyond that tolerance zone and it cannot make corrections to bring itself back onto plan, the vehicle will ask its platoon leader for a new plan.

The individual vehicle receives its plans from the platoon and in turn performs behavior generation tasks to refine the plans for its subordinates in order to achieve its orders. The vehicle's subordinates are the subsystems that reside on it, such as locomotion, attention, communications, and mission packages. The locomotion subsystem controls the movement of the vehicle, attention controls the vehicle sensors for mobility, and communications controls all tasks pertaining to transmitting and receiving. Mission packages will vary, but include reconnaissance, surveillance, and target acquisition (RSTA). The vehicle commander assigns jobs to its subsystems, possibly in collaboration with the subsystem commanders. The vehicle-level world model contains names and attributes of objects, such as size and shape of obstacles. A map or grid-based representation of the local terrain is used for local navigation and sensor pointing. The sensory processing module, which extracts object dimensions, locations, and relative velocity, populates this map. The local map covers about 1-kilometer range with a resolution of 30 cm and may be merged with an *a priori* digital terrain map downloaded from the OCU. Value judgment evaluates candidate plans for effectiveness of meeting mission goals and sensor dwell times. Local planning uses sensor data to avoid obstacles and honor other requirements, such as stealthiness, or maintaining radio contact with the OCU. Schedules of activities for the subsystems are generated for 3 s intervals, out to a planning horizon of about 30 s.

The 4-D/RCS hierarchy levels below vehicle are the subsystem, primitive, and servo levels. Each of these levels is built from compute nodes containing behavior generation, sensory processing, world modeling, and value judgment functional elements. At each level, planning occurs to achieve the commands and goals received from the superior level. The timing and spatial horizons grow smaller, by about a factor of 10 with each lower level. The granularity of the data is smaller, also by roughly a factor of 10. Therefore, the computational complexity of each level of the 4-D/RCS hierarchy remains consistent.

The functions corresponding to platoon leader and vehicle commander can reside on the OCU or on a vehicle. Only the Tier 2 OCU will be capable of generating and controlling platoon-level (multi-vehicle) plans. Both Tier 1 and Tier 2 will be able to generate and control single vehicle plans. Both Tier 1 and Tier 2 will be able to monitor vehicle progress, regardless of whether it is part of a platoon mission or working alone. When necessary, the vehicle's surrogate platoon leader function can be activated on a vehicle designated to have command of the platoon. Selection of the lead vehicle may either be done at the OCU at the start of the mission, or may be negotiated among the vehicles when required. Since the OCU may be connected to battlefield sensors, it may have a much more complete and up-to-date picture of the battlefield than the lead vehicle. This increased world knowledge makes the OCU the most desirable module to generate the overall platoon-level plan. However, if all of the vehicles are out of radio contact with the OCU, the lead vehicle is capable of generating a new platoon-level plan from its surrogate platoon planning level using its existing world model databases. When the group of vehicles reestablishes contact with the OCU, this new platoon-level plan will be sent back for review. Vehicles that are out in the field capture more up-to-date information about local conditions and may share these with their peers, leader, and OCU. For instance, a vehicle may detect that a road that the platoon was expected to cross is impassable, necessitating re-planning for current and future military units traversing the area. Replanning may also be initiated by the user via the OCU, as a result of mission rehearsal, mission monitoring, or receipt of a new mission.

Irrespective of where the platoon and vehicle planner reside, the same behavior generation, world modeling, value judgment, and sensory processing algorithms must be developed and exercised. The world model and sensory processing functional elements will have access to different information, depending on whether they reside on the OCU or on a vehicle. However, they will apply the same processes. Decomposing military orders into a set of tasks, waypoints, and control parameters for each level of the hierarchy requires a variety of algorithmic approaches. If a platoon is given an order to perform a road march, several functions are involved in the behavior generation. Selecting the platoon's marching formation can be achieved using a simple rule-based approach based on military doctrine. A graph-based search algorithm, such as A*[3], is used to generate paths for the vehicles. The path generation algorithms run on a discretized representation of the terrain and conditions relevant to the mission. Cost maps must be generated that reflect the positive or negative values (or benefit versus cost) for each segment to be potentially traversed. Terrain elevation and trafficability, inter-visibility, location of enemy positions, vehicle capabilities, and other information are used to build the cost maps. The costs are tuned to the primary military objectives for the mission, such as stealthiness being more important than speed. A behavior generation functional element creates alternative plans that are evaluated by the value judgment functional element for potential of success. The platoon or vehicle planner can also invoke "canned" behaviors. For instance, if the vehicle becomes stuck in mud, it may start a pre-planned set of actions to try and free itself.

Vehicle to vehicle communication is necessary to allow the vehicles to maintain military road march formations, perform coordinated activities, and operate when communications to the OCU is limited. For example, when the vehicles are performing a road march a route plan will be calculated for the entire platoon. This platoon-level plan will be downloaded to all of the vehicles with the vehicles negotiating among themselves for leader/member relationship, and position in the formation. Once the mission has commenced, the lead vehicle will broadcast its position to all of the member vehicles to allow them to maintain tight formation. The member vehicles will monitor the lead vehicle's status and position through Global Positioning Satellites (GPS), and space themselves according to their required position in the formation.

An example of coordinated vehicle activity would be the case of target hand-off for RSTA applications. In this scenario, one of the vehicles would detect a potential threat moving through its RSTA sensor's field of view. The vehicle would contact other robotic vehicles to either confirm the threat or continue tracking the threat once it has left its field of regard. In the case when communications to the OCU is limited, it is possible that communications to an individual vehicle may be routed through other vehicles.

OPERATOR PLANNING INTERFACE

The Tier 1 OCU has a simple interface to allow the operator to manually enter vehicle plan information. Waypoints are entered in Latitude and Longitude, or relative to the current location. When the Tier 2 OCU is being utilized, the operator interface has more capabilities. Based on the type of mission selected by the user, the user will graphically select:

- goal points (route reconnoitering), or the area to be reconnoitered (area reconnoitering or bounding over-watch),
- waypoints along the route,
- the number of vehicles available to perform the mission, and
- other mission parameters/constraints (e.g., time of mission, enemy coverage (both sensor and weapon) concealment, driving conditions, etc.).

The planner will then formulate the platoon-level plan based on the data input by the user and by data existing in the available databases. This plan is then downloaded to the vehicles for execution.

SUMMARY

This paper described the proposed MMI/MP software systems for implementation in the Demo III XUV program. The system consists of technologies in the areas of User Interfaces, Mission Planning, and Real-Time Map Database Terrain Fusion. The end product of this effort will be the control of a platoon of robotic vehicles that are capable of carrying out a scout-type mission in an autonomous mode. Much has already been accomplished in this program, including an MMI that consists of a two-tier approach for scalable control of the vehicle systems. The Tier 1 system provides control via a web based Java application executing on a low-end system. The Tier 2 OCU provides an integrated automatic mission planning system for all vehicles in the platoon in addition to a 2D and 3D status display.

The system architecture consists of a six level planning system based on the 4-D/RCS architecture developed by NIST. The system has two high level planners which execute on the OCU (or on the vehicle when the OCU is unavailable), and a set of four planners which execute on the vehicles in a hierarchical 4-D/RCS implementation.

---

[1] J. Albus, 4-D/RCS: A Reference Model Architecture for Demo III, NISTIR 5995, 1997.
[2] Koller, D., Lindstrom, P., Ribarsky, W., Hodges, L. F., Faust, N., and Turner, G., "Virtual GIS: A Real-Time 3D Geographic Information System", *College of Computing and Graphics, Visualization & Usability Center*, Georgia Institute of Technology
[3] Nilsson, N.J., "Principles of Artificial Intelligence", Tioga Publishing Company, 1980.