Representing Ground Robotic Systems in Battlefield Simulations

M. Fields
U.S. Army Research Laboratory
Aberdeen Proving Ground, Maryland 21005-5066
mafld@arl.army.mil

ABSTRACT

As the Army continues to develop robotic systems for combat and combat support missions, it needs to also develop representations of intelligent system performance for its battlefield simulation tools. These simulation tools differ considerably in their level of abstraction, flexibility, and scale. Constructing the actual performance model requires the modeler to consider three factors: 1) the purpose of the particular simulation study, 2) the overall fidelity of the target simulation tool, and 3) the elements of the robotic system that are relevant to the simulation study. In this paper, we discuss a framework for modeling robotic system performance in the context of a battlefield simulation tool. We apply this framework to a model of the Demo III robotic system used in the OneSAF simulation tool.

1. INTRODUCTION

As the U.S. Army continues to develop concepts such as the Future Combat System (FCS) which include robotic assets, it needs to develop representations of intelligent systems for its battlefield simulation tools. This is a formidable task. Robotics systems currently being developed range from human-portable systems to large tracked or wheeled vehicles. The level of control required for these intelligent systems ranges from full-time remote operation to intermittent supervisory control. The simulation tools themselves have different levels of fidelity, different time scales, and different intended uses. These tools allow the technology developers, the analysts and the soldiers to experiment with robotic systems in readily available, re-configurable virtual environments. Technology developers can use simulations to investigate system design questions such as payload composition and placement, vulnerability, and the appropriate sensor mix for autonomous mobility. The soldiers and military analysts can use simulations to develop Tactics, Techniques and Procedures (TTP) and requirements for robotic systems based on parametric studies involving key scenarios run over several terrain databases representative of the types of environments the robot is likely to encounter. Finally, well-designed simulations can be used to identify critical near term technology problems and help prioritize research efforts.

The purpose of this paper is to present a framework for modeling intelligent systems which applies to a wide range of battlefield simulation tools and simulation purposes. Table 1 shows a breakdown of the types of models and simulations used to support weapon systems development and acquisition. The table gives a level of detail for the model and some examples of the types of evaluations and model output that can be expected at each level. In general, models that fall in categories near the top of the table represent systems more completely than simulations in categories near the bottom of the table. Traveling down the table, the size of the simulated world and the number of entities represented in a battlefield engagement increases. The categories are somewhat artificial; there are models and simulations that fall somewhere between categories given in the table. Two of the battlefield simulation tools currently being used to examine robotic systems are the Combined

Simulation Category	Level of Detail Modeled	Performance Data/Models Required	Type of Evaluation	Example Output
First Principal Physics	Physical processes	*Not applicable*	Design Feasibility	Electric Field Strength
Engineering	Components, Subsystems	Possibly Subcomponent level	Subsystem Performance	LADAR elevation map
One-on-One	Complete Weapon Systems	Component level	System Performance	Probability of successfully navigating a cross-country path
Few-on-Few	Small Military Units (Squads to Company)	Component level System level	System Effectiveness	Specific Exchange Ratio (SER) *Red losses caused by a specific blue system*
Force-on-Force	Large Scale Combat Unit (Battalion or Higher)	System level	Combat Utility	Loss Exchange Ratio (LER) *Ratio of red to blue losses*

Table 1 A hierarchy modeling and simulation tools used to support weapons systems development and

Report Documentation Page

Form Approved
OMB No. 0704-0188

Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.

1. REPORT DATE **AUG 2002**	2. REPORT TYPE	3. DATES COVERED **00-00-2002 to 00-00-2002**

4. TITLE AND SUBTITLE **Representing Ground Robotic Systems in Battlefield Simulations**	5a. CONTRACT NUMBER
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **U.S. Army Research Laboratory,Aberdeen Proving Ground,MD,21005-5066**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES
Proceedings of the 2002 Performance Metrics for Intelligent Systems Workshop (PerMIS -02), Gaithersburg, MD on August 13-15, 2002

14. ABSTRACT
see report

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**Same as Report (SAR)**	**6**	

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std Z39-18

Arms and Task Force Evaluation Model (CASTFOREM), and OneSAF. OneSAF has been used to support the Demo III robotics program. CASTFOREM will be used to provide weapon systems analysis for the FCS program.

The U.S. Army Training and Doctrine Command uses CASTFOREM to study force composition and system effectiveness at the brigade and battalion level. It is primarily a force-on-force event-driven simulation. Processes such as detection and individual system damage are modeled stochastically using performance data provided for each weapon system. The actions of combat units are controlled by expert systems. Human participation is limited to preparing the data, rule sets for the expert systems, and scenario design [1].

OneSAF is a real-time distributed interactive simulation tool developed by the U.S Army Simulation, Training and Instrumentation Command. It is used to training soldiers and to examine weapon systems concepts in brigade and below scenarios. It can be used to model engagements ranging in size from one-on-one encounter to battalion level exercises so it spans several of the categories given in Table 1. The actions of individual or aggregate units are controlled by behavior algorithms or human participants. Since it is a distributed simulation, it can be used in exercises involving different types of simulations, simulators, and actual systems [2].

Constructing a particular robotic system model requires the modeler to consider three factors. First, it is important to keep in mind the purpose of a particular simulation study. Examining the contributions of a robotic scout to a battalion-level movement-to-contact scenario requires a much different model than examining the effect of a planning algorithm on autonomous driving. The overall fidelity of the model is also a major consideration. Higher fidelity simulation tools are compatible with physics-based models of robotic systems and subcomponents. Lower fidelity simulation tools use simple mathematical functions, often given as lookup tables, to represent subsystem performance. The quality of these lookup tables depends on the experimental data that can be collected for the robotic system being modeled. Finally, in constructing a model of the robotic system, a modeler needs to consider the elements of the robotic system that are relevant to the study. For instance, the overall performance of the driving sensor suite is certainly important for evaluating the contribution of robotic systems to a scout mission. The performance of an individual driving sensor may be less relevant.

In the next section, we present a general framework for robotic models, which identifies the critical elements of a robotic system that need to be represented in any battlefield simulation. In the third section, we present some of the modeling and simulation tools that have been developed to support the Demo III robotics program. We also discuss how these tools can be used to guide the development of the robotic system performance models required by the Force-on-Force models such as CASTFOREM.

2. A SYSTEM-LEVEL DESCRIPTION OF A ROBOTIC SYSTEM

It is useful to take a systems engineering approach and define a simulated robotic system as a collection of interlinked subsystems. It is important to note that this definition of the robotic system includes both the robotic vehicle(s) and the operator. Robots (even autonomous systems) cannot operate for long periods of time without human intervention. In terms of a battlefield scenario, the operator receives a mission and employs his robotic assets to complete the mission. The diagram given in Figure 1 shows a notional robotic system consisting of five major subsystems: Navigation; Communications; External Command and Control; Internal Command and Control; and the Payload System. Each of the major subsystems has elements relating to the mechanical and software components of the system.

In this notional robotic system, the External Command and Control System consists of the human operator, the human-machine interface, and any command decision aids

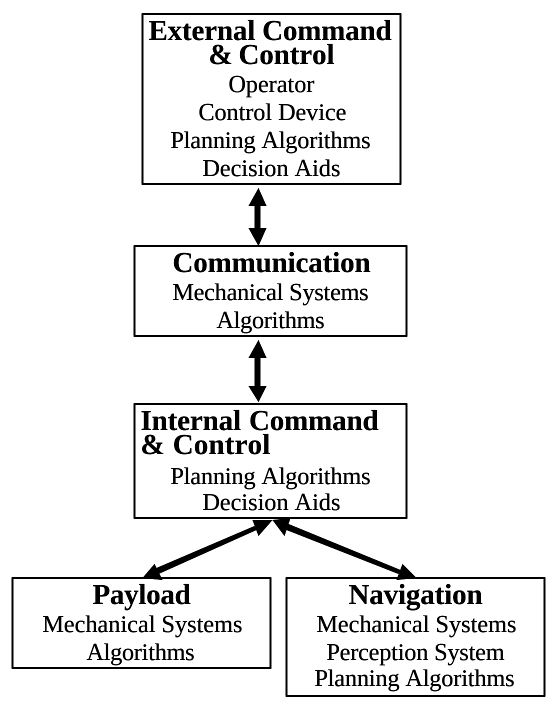

Figure 1. A systems engineering representation of a sample robotic system

the operator may use. Depending on the application, this system could be situated near the robotic vehicle or much farther away. The Navigation System contains sensors and other hardware and software such as perception and planning algorithms. It resides on the robotic vehicle. The Communication System consists of the radios that link the human operator to the robot and the associated software. Components of this system are on the robotic vehicle and are also co-located with the External Command and Control system. In one sense, the Navigation and Communications systems are support systems intended to allow the payload system on the robotic vehicle to contribute to the tactical mission. The Demo III robot carried a Reconnaissance Surveillance Target Acquisition (RSTA) package; other payloads such as weapons, storage containers, or smoke generators could also be represented. In general terms, the composition of the payload system consists of mechanical systems and supporting software algorithms.

The arrows in the pictures indicate the data flow in the system. The operator uses the communication system to give commands to the subsystems on the robotic vehicle. In this notional system, commands are passed through an intermediary command and control system that resides on the robot itself. All the subsystems interact with the simulated exercise, at least to some degree. They are subject to degradation or damage from elements in the simulated environment or from other entities in the exercise. Some systems such as the navigation system gather information from the environment to be used by systems on board the robot or at the external command and control station.

Diagrams similar to Figure 1 describe most weapon systems; we are using it to illustrate the types of models and supporting data needed to represent robotic systems in a battlefield simulation. There are two distinctions between robotic systems and most other weapons systems. First, mobility system performance must consider not only chassis response but driver reliability as well. Second, the robotic system is distributed. Many processes are semi-autonomous, requiring some level of operator participation. In a robotic system, time-dependent performance measures such as "average target identification time" include communication time and two types of processing time (robotic processing time and human processing time).

For any study, we need to represent the performance of the major systems shown in Figure 1. Within engineering level simulations, the major systems may be represented by collections of high fidelity models of each of the processes contained within the system. Another engineering or quasi-engineering approach is to embed system components into a simulation tool. This allows researchers to "virtually" test hardware and software during the development process. As simulations become more abstracted, less detail can be included in the performance models. As researchers construct abstracted models of each of the systems, it is important to keep the following questions in mind. First, what is the purpose of the system? How reliably does the system accomplish its purpose? Finally, how quickly does it accomplish its purpose? The speed and reliability questions depend on collecting and analyzing experimental data

A model of command and control must consider the types of decisions the operator makes, the speed of the decision process and the reliability of the decision process. Right now, data about the performance of the decision process are sparse. We can collect data from "virtual" exercises using embedded decision software or from field exercises.

To represent communications between the operator and the robot in a large-scale simulation, we need to measure the size and frequency of the messages. We also need to measure the speed and reliability of the system. We can gather some of this information from high fidelity communication models. Most of the information should come from integrated field experiments, where the robotic system has to accomplish mission similar to those used in combat.

Representing the navigation and payload systems also depends on gathering data to determine the speed and reliability of the process.

In the next section, we describe some of our current models of the Demo III robotic system. Since the emphasis of the Demo III program is on developing autonomous mobility technologies, most of our modeling efforts have been directed at autonomous mobility as well. Many of the models we have developed describe processes within the navigation system. Recently, we have begun developing models for the other systems as well.

3. MODELING THE DEMO III ROBOTIC SYSTEM

Under the Demo III robotics program, the U.S. Army is developing a small survivable experimental unmanned ground platform (XUV) capable of autonomous operation on rugged terrain. Although the primary focus of the Demo III program is to develop and demonstrate autonomous mobility technologies, the research was focused on providing a robotic system for platoon-level scout missions.

The Demo III XUV was designed in accordance to the NIST Real Time Control (4D/RCS) Reference Model Architecture which is a hierarchical structure designed to support the development of autonomous systems. Each level in the hierarchy is referred to as a node. A node consists of a behavior generation element, a value judgment element, a world model element, a sensory processing element and a knowledge database. The level of detail and dimensions of the "world" in the world model is a function of the node's position in the hierarchy; a node controlling several vehicles needs less resolution over a larger region to plan than a node that controls a single vehicle. Nodes receive goals, priorities, and plans from superiors and produce goals, priorities, and plans for subordinates.

The five levels of the 4D/RCS architecture are Section Level, Vehicle Level, Subsystem Level, Primitive Level, and Servo Level. The section level receives a general plan generated at a higher level such at the platoon level. This plan contains a general command such as "Conduct a Tactical Movement" and a plan based on *a priori* information such as digital maps and situational awareness overlays. A section level plan is generally used to control multiple robots. At the Vehicle Level, the vehicle refines the Section Level command by developing a plan based on its world model which contains digital map data, situational awareness information and low-resolution information gathered by the on-board sensors. At this level, the vehicle refines the Section Level plan to avoid relatively large problem areas. At the Subsystem Level, the robot plans paths to avoid obstacles in its path. The Primitive Level controls the steering, acceleration, and braking of the robot. The lowest level in the architecture is the Servo Level; it controls the actuators for each of the subsystems.

Most of the modeling work for the Demo III system has focused on small scale battlefield experiments and engineering level studies. The primary purpose of these models has been to support the system design process. Systems are represented by collections of models for each of the major systems given in Figure 1. As the technology matures, the community can begin to develop system level performance models. The challenge is to capture the system characteristics contained in the current collection of engineering and quasi-engineering level models into one model or mathematical function representing each of major systems. In the rest of this section, we discuss some of the existing OneSAF models representing processes and sub-processes within the major systems. We are beginning to examine performance models

One of the modeling and simulation tools used to support the Demo III robotics program is OneSAF. It is a real-time battlefield simulation tool with a time step of ~0.015 seconds (66 Hz). It is an entity-level simulation so all units are represented by collections of individual soldiers or vehicles. The baseline OneSAF represents hundreds of U.S. and foreign weapon systems and units. It has many pre-programmed behaviors to control the movement and interactions of those systems.

OneSAF is suitable for studying the interaction of robotic systems with other systems participating in small battlefield engagements. Because it is designed to interact with human participants, OneSAF is also appropriate for developing potential TTP for the use of robotic systems. However, because of it is a real-time simulation, it may not be appropriate for parametric studies requiring several replications. OneSAF's time step and battlefield environment are also too coarse for most engineering level studies. For example, its environment is not detailed enough to be useful in evaluating the driving sensors or the perception algorithms involved in autonomous mobility. However, it can be used as a quasi-engineering simulation tool for aspects of autonomous mobility such as vehicle level planning that have a relatively long cycle time. We discuss this application further in Section 3.1.

3.1 Autonomous Mobility

The autonomous mobility model for the Demo III robotic system consists of the three main elements – a movement equation, a sensory processing suite, and a planning suite. The movement equation is a simple point model that determines the position, velocity, and acceleration of the vehicle at the end of each time step. The sensory processing suite builds a world model from inputs provided by the driving sensor suite. The planning suite uses the world model to determine a suitable path for the robotic vehicle. In the next couple of paragraphs, we describe the models that we used to represent each of these elements. In general, we can relate our modeling strategy to the 4D/RCS architecture. We can represent many of the processes at the Subsystem, Vehicle, and Section levels as algorithms that are executed in real time as a part of the overall simulation. However, We must depend on data and mathematical abstractions to represent processes on the Servo Level.

The time step for OneSAF is approximately 0.067 second. In this amount of time, the robot travels less than 1 meter (The maximum speed for the XUV is 40 kilometers per hours). We could excite the movement equation with sub-meter resolution terrain. Some high fidelity terrain databases for OneSAF are available, but they require large amounts of computer memory to use them efficiently. In our research, we use primarily 100 m and 30 m resolution terrain databases. We use a relatively simple equation of motion to model the motion of the XUV that uses the current position, velocity, acceleration and desired direction as input and gives the new position, velocity and acceleration as output. This equation is used in OneSAF to describe the motion of many of the ground vehicles.

Building a world model of the environment requires the driving sensor suite to gather information from the environment, process it, and present it to the planning suite in the form of a world model. The time step in OneSAF does not permit us to model the activities of the sensors themselves. Instead, we model the process of generating the world model from the simulated terrain database. In our simulation studies, we want the robotic vehicles to respond to relatively small obstacles such as woody vegetation and ditches that are not available on the *a priori* map. These are not features of a typical OneSAF terrain database. In our prior research [4], we developed techniques to add these features to existing OneSAF terrain databases. Figure 2 shows a section of a OneSAF terrain database with two types of mobility obstacles positive obstacles shown dark gray and negative obstacles shown in light gray. Each of these obstacles is a polygonal feature with associated parameters used to specify a probability of detection function. Right now, detection of a particular obstacle depends on the type of the obstacle, its size (length, width, height), and the distance from the vehicle.

We produce two types of world models from the terrain database information. The first type of world model is a two-dimensional obstacle map with three types of pixels (clear, unknown, and blocked). Unknown pixels indicate areas within driving sensor range that are blocked from line of sight. Blocked pixels show the location of detected obstacles. Clear pixels indicate regions of the terrain that are visible to the driving sensor suite and are free from detected obstacles. This is a useful representation of the obstacle detection process, but it is not the best representation of the world model used by the XUV. The XUV uses an elevation map to plan its near-term movements. Figure 2 shows a two-dimensional obstacle map in the context of a large battlefield map. In the obstacle map, green indicates clear areas, yellow indicates blocked areas, and gray indicates unknown areas. An elevation map can be produced from the same data. The heights in the elevation map are derived from two sources. The terrain skin provides the underlying ground plane elevation; detected obstacle polygons add or subtract elevation from this ground plane.

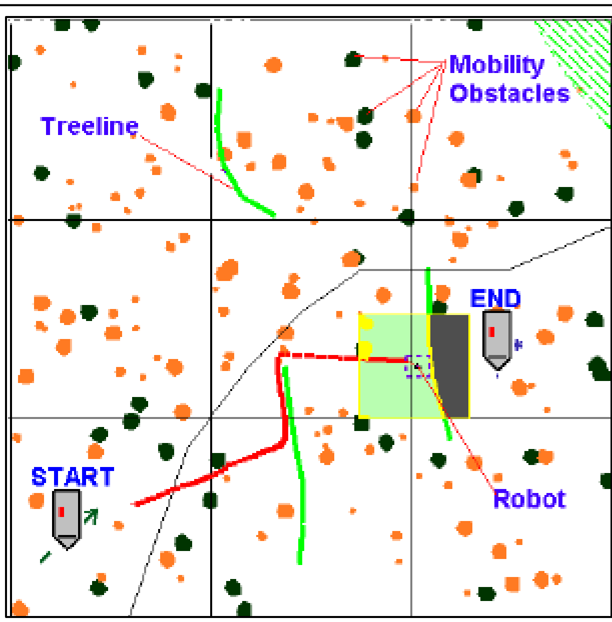

Figure 2. A battlefield snapshot showing the robot and its obstacle map

The planning process on board the vehicle consists of two planners: a near-term planner operating at the subsystem level of the 4D/RCS architecture and a mid-range planner operating at the vehicle level of the architecture. In our work, we have developed two different models of the Demo III robotic planning process. In collaboration with the National Institutes of Standards and Technology (NIST) and Science and Engineering Services, Inc (SESI), we developed one model designed to examine the performance of the actual robotic planning software in tactical missions. This model requires software components internal and external to the OneSAF simulation code. The actual vehicle level planner was linked to the OneSAF simulation code using the NIST neutral message language (NML) to pass plans from the planner to the simulated XUV. World models were passed from the simulated entity to the planner allowing it to use information gathered by the driving sensors on the simulated robot.

This same technique of linking actual software to the simulation system can be used to include the near-term planning system. In this case, the three-dimensional elevation map is passed to the near-term planner from the simulated world. Paths are passed back to the simulated entity.

The linked simulation is a good method to gather data about planning algorithm performance and to support the algorithm development process. We can experiment with the planners in different situations varying the tactical situation and the obstacle distributions.

In the context of a larger exercise, possibly using another simulation tool, it may be impractical to link the actual code with the simulation. In this case, we want to use surrogate algorithms or mathematical models that perform similarly to the actual planning algorithms. We have used simpler algorithms to represent the near-term planning process. These algorithms use the two-dimensional obstacle map to plan the path of the vehicle.

3.2 The External Command and Control

The external command system for the Demo III robotic system consists of the operator, the operator control unit (OCU), and the associate planning software. There are two ways to represent the external command and control. The first method is to put a human operator in the simulation loop. The Mounted Maneuver Battlelab and SESI used this approach to support the Demo III program. The OCU was linked to OneSAF via NML to pass plans and other information between the operator and the simulated robotic entity. This approach of embedding hardware and software components into a simulation study allows researchers to collect data about operator activity and workload. Such information can be used to guide the design of effective control devices. Information from the embedded model also provides some system performance information that can be used to construct performance models for complex battlefield simulations.

We are beginning to construct an abstract model of the human operator. In its simplest terms, the human operator controlling one or more robotic assets is a server with a queue of heterogeneous tasks to service. As with any queuing problem, it is the frequency and service times for each type of task that determines the workload on the operator. In our model, there are two types of service requests: mobility assistance requests and RSTA assistance requests.

3.3 The Communication System

We are beginning to address communication system models. Our approach is to model the amount of time required to transmit a message between the robot and the operator based as a function of message size. We are using this model in connection with our queuing theory model of the operator to introduce delays into requests for service and operator response time.

3.4 The RSTA Payload System

The RSTA system model uses existing models from the OneSAF simulation package. These models can represent many systems, including camera systems, forward looking infrared devices, and radar systems.

4. CONCLUSIONS

In this paper, we presented a framework for developing models of intelligent systems which applies to a wide range of battlefield simulation tools and simulation purposes. The framework consists of five major systems: External Command and Control; Communications, Internal Command and Control, Navigation and Payload. Each of these systems needs to be represented in a battlefield simulation, regardless of the level of simulation. In lower level simulations, we are able to use detailed models and/or components of the robotic systems to represent the robotic system. As the scale of the combat model increases, we need to develop abstract performance models of the systems within the robotic system. The validity of these performance models depends on the experimental data used to construct them.

5. REFERENECES

[1] Army Modeling and Simulation Resource Library .http://www.msrr.army.mil/index.cfm?RID=MNS_A_1000185, 1998.

[2] R. Witman and C. Harrison, "OneSAF: A Product Line Appraoch to Simulation", Technical Report, The Miter Corporation, 2001

[3] J. Albus,"4D/RCS Reference Model Architecture for Unmanned Ground Vehicles", Proceedings of the SPIE Vol. 3693 AeroSense Session on Unmanned Ground Vehicle Technology, Orlando FL, April 7-8, 1999.

[4] M. Fields, "Modifying ModSAF terrain Databases to Support the Evaluation of Small Weapons Platforms in Tactical Scenarios", ARL-TR-1996, Army Research Laboratory, Aberdeen, Maryland, August 1999.