ADAPTING A DISTRIBUTED REPRESENTATION SYSTEM TO THE DISTRIBUTED SIMULATION ENVIRONMENT

Alan R. Dickens
BBN Systems and Technologies
Cambridge, Massachusetts

ABSTRACT

The Simulation Networking (SIMNET) program demonstrated the feasibility of conducting tactical training and combat development in a distributed simulation environment. As other programs build on the success of SIMNET, the number of types of vehicles simulated will continue to increase dramatically. In a distributed interactive simulation system, every vehicle is displayed by other manned vehicle simulators in several ways: visually, aurally, and with infrared, radar, sonar, and other sensors. Currently, every manned simulator must keep a set of representations for every other type of vehicle. Unfortunately, this means that whenever a new type of vehicle is added to the network, all other entities on the network must be updated to include representations of the new vehicle.

Alternatively, each simulator could bring with it a generic representation of itself which could be distributed to other entities on the network. Such an approach is referred to as a distributed representation system. This paper discusses a set of extensions to the Distributed Interactive Simulation (DIS) protocols that would be necessary to implement such a system.

ABOUT THE AUTHOR

Alan Dickens is a Software Engineer with BBN Systems and Technologies. He has seven years experience developing distributed vehicle simulation. One of the original engineers on the SIMNET project, he has developed software that models hull and turret kinematics, damage and failures, and fire control systems. He was also involved in the development and implementation of the original SIMNET protocols. Recently, he has participated in the systems engineering for several follow-on distributed interactive simulations. Mr. Dickens holds a BA in Computer Science from Brown University.

ADAPTING A DISTRIBUTED REPRESENTATION SYSTEM TO THE DISTRIBUTED SIMULATION ENVIRONMENT

Alan R. Dickens
BBN Systems and Technologies
Cambridge, Massachusetts

INTRODUCTION

The SIMNET program developed the concepts necessary to construct a distributed interactive simulation environment. With the growing acceptance of the DIS standards,[1] an increasing number of new types of vehicle simulators will be designed to interact in the simulated world. These simulators will be built by a variety of manufacturers, and will be based on different types of hardware.

As the simulated world expands, a number of problems will likely arise. Some of these problems will relate to the increase in the number of simulated vehicle types, and to differences in vehicle representation by manned vehicle simulators made by different manufacturers. Many of the problems can be addressed by allowing every vehicle to maintain and distribute models representing its own appearances.

This paper presents a background of current distributed interactive simulation applications with particular emphasis on issues relating to vehicle representation. After discussing the requirements of specific types of vehicle representations, it then addresses the problems with current mechanisms in more detail. Finally, a solution to some of these problems is presented, along with a summary of issues requiring additional investigation.

BACKGROUND

SIMNET was an advanced research project sponsored by the Defense Advanced Research Projects Agency (DARPA) in partnership with the United States Army. The program developed the technology to build a large-scale network of interactive combat simulators.[2] Subsequent programs continue to increase the number of types of distributed simulation entities, which include both manned vehicle simulators and computer-controlled Semi-Automated Forces (SAF) (see Figure 1). This paper assumes a basic familiarity with the concepts of distributed interactive simulation; for a more thorough discussion, refer to "The SIMNET Architecture for Distributed Interactive Simulation", by Duncan C. Miller.[3]

After defining some terms, a list of types of representations is provided, noting aspects that are relevant to the problem of distributed representation. This is followed by a discussion of the common applications of distributed simulation, again noting aspects that are relevant to the representation issue. Finally, a detailed discussion of the problems with the current representation scheme will lead into a discussion of distributed representation.

Terminology

For the purposes of this paper, a "simulated vehicle" is a single tank, aircraft, or even a soldier. Every simulated vehicle's representation must be modified based on the condition of the vehicle. Modifiers may include burning, smoking, or catastrophically killed, as shown in Figure 1. A "simulation host" refers to the physical equipment that is capable of generating one or more simulated vehicles. For example, a manned vehicle simulator is a single simulation host, which normally generates a single simulated vehicle. A SAF workstation, while also a single simulation host, is capable of generating many simulated vehicles. A "differentiated part" is any component of a vehicle's representation that must be dynamically updated. A typical example of a differentiated part is a tank turret. An "effect" is the presentation of any event-based occurrence. Examples of effects include weapons fire and impact, tracers and missile models, dust and vapor trails, etc.

Types of Representations

A simulated vehicle is represented by other manned vehicle simulators in a variety of ways.

<u>Visual.</u> The vehicle is represented as a set of polygons that usually have both color and texture maps associated with them (see Figure 2). Differentiated parts might describe the orientation of a turret or gun. Effects that might be generated by the vehicle are also part of the representation,

such as a dust cloud trailing behind a moving vehicle, or a gun muzzle flash. A vehicle may also have a set of markings associated with it (e.g., bumper numbers) that identify that vehicle to those who can decipher the markings.

Infrared. This is similar to the visual representation, but is based on the emission of heat rather than the reflection of light (see Figure 3). While the polygons may be the same as those in the visual model, the colors and texture maps will certainly be different. Additionally, since the infrared representation depends on heat, the simulated entity must be able to tell displaying vehicles how hot it is. Differentiated parts could describe the "hotness" of arbitrary sets of vehicle components (gun barrel, engine, tracks, surfaces exposed to direct sunlight, etc.).

Aural. This is achieved through the external sounds that a vehicle makes. Although many of the sounds that a vehicle simulator makes are internal and are intended solely for the vehicle's crew, some sounds originate externally, particularly the weapons effects from other vehicles. Aural cues would also come from differentiated parts. For example, the sound of a vehicle passing near by might be parameterized by how fast it is moving.

By reflection. Here, a set of representations relies on the reflection of various wavelengths (radar or sonar). These representations are generated based on the interaction between the emitting source, the shape and construction of the vehicle being represented, and the capabilities of the receiver.

Applications of Distributed Simulation

Currently, distributed simulation is used for three purposes: training, combat development, and large-scale simulation. Each of these functions places a different set of requirements on the hardware, software, and network protocols used. This discussion assumes a basic understanding of how distributed vehicle simulators are used, and will highlight certain characteristics that have special relevance for representation issues.

The original SIMNET system was designed to support a force-on-force tactical training application. A training center's operations are the most routine of any distributed simulation application. Troops are brought in to run through a generally established set of scenarios. Their opponents are frequently SAF, but sometimes they train against other manned simulators. While a training environment will have a large number of manned vehicle simulators, only a few vehicle types are typically available. In this environment, the set of vehicle types that interact on the battlefield is well known, so it is quite straightforward for every vehicle to know about every other vehicle's representations.

One disadvantage to training simulators is that since the hardware is replicated many times over, economic considerations often limit the availability of spare computational resources. This increases the difficulty of maintaining compatibility with newer systems. Also, despite their relatively routine day-to-day usage, training simulators do occasionally encounter new types of vehicles when they are linked to other sites via a Long Haul Network (LHN).

Combat development applications tend to involve many fewer vehicle simulators. A typical application involves a handful of vehicle simulators conducting the same exercises many different times to obtain statistically significant data. To allow the repetition of scenarios, the opponent is almost always SAF. In contrast to training applications, combat development experiments must involve new and different vehicle types. In fact, they frequently involve notional vehicle types that are being tested for battlefield effectiveness. Combat development simulators are usually short-lived, so generational compatibility is not an issue. Also, since they usually require special functionality and are built only in small numbers, combat development vehicle simulators usually utilize the newest hardware and software, and so tend not to be computing resource limited.

The third current use of distributed simulation involves leveraging distributed simulation technology in the large-scale wargaming arena. This can be a battle reenactment, or it can be a large-scale proof of principal, involving many different types of vehicles from geographically distant sites.

Increasingly, large-scale exercises involve interaction among vehicle simulators built by different manufacturers. For example, the 1992 I/ITSC Conference is planning to demonstrate the interoperability of simulators made by over two dozen different manufacturers. Because of the heterogeneous nature of the participating network entities, a large-scale exercise must be able to accommodate simulators of greatly varying capabilities (see Figure 4).

Current Representation Problems

In SIMNET, each simulation host contains a complete set of representations for every vehicle that it might encounter. These representations are generally in a format specific to the hardware used for the simulator, which leads to several problems. First, any time a new vehicle is added to the simulated world, every other simulation host must have its data files upgraded to include representations of the new vehicle. Second, if the existing representations for a vehicle are to be used with a simulator that uses different hardware, the representations must be converted or remade to run on the new hardware.

Additionally, the SIMNET and DIS protocols limit the expression of differentiated parts, restricting the ability of simulated entities to increase the fidelity with which they are represented. In SIMNET, for example, infrared images of other vehicles are static, in that they do not depend on the amount of heat (or lack of heat) that the vehicle is generating. The image could easily be based on such parameters as vehicle speed or whether the engine is running, since this information is readily available. It would be much more difficult, however, to base the image on the accumulated heat within the vehicle, since this is information that only the vehicle itself can reasonably be expected to keep track of, and this information is not currently included in the update packets that are broadcast to observing entities.

Another problem is the difficulty of maintaining the forward compatibility of older vehicle simulators. Typically, they load the entire set of known vehicle representations into memory on startup. In some cases, the amount of memory available is the limiting factor on how many vehicle representations can be included in this set. This practice is wasteful, since only a subset of those representations is normally used during a given exercise.

DISTRIBUTED REPRESENTATION SYSTEMS

A working distributed representation system would overcome these problems. In such a system, every simulation host would have sole responsibility for maintaining the representations of all entities that it simulates. This would include both vehicle models and event-based effects. This information would be maintained in a generic format that would be accessible to all network entities in an exercise.

Protocol Implications

Several protocol changes would be required to implement a distributed representation system. First, and most obvious, would be the definition of a standard format in which the representations could be transmitted. USAF Project 2851 is working to produce a database interchange format specification that would codify a generic system of visual and infrared representation.[4] A similar standard for aural representations would also be necessary.

In addition to defining the format for various types of representations, different vehicle states must also be considered. A given type of vehicle is likely to have several different states, including live, burning, smoking, and charred. All of these states must be accounted for in the model representations.

One of the most difficult problems in a distributed representation system is representing attenuation over distance. For sounds, it is fairly straightforward to attenuate the representation over distance. For electromagnetic phenomena, propagation models have already been developed which may prove sufficient.[5] For visual or sensor images, however, this is somewhat more problematic. Visual systems typically use different level-of-detail (LOD) models to attenuate the representations of vehicles in the distance. The choice of LOD model is based on the range to the model and the resolution of the view in which the model is displayed. Therefore, it might be sufficient to transfer multiple LOD models as part of the representation, and then allow Computer Image Generators (CIGs) of different resolutions to choose when to display which LOD model. Alternatively, a mechanism could be developed by which lower LOD models are automatically generated from the highest LOD model. Obviously, this is a complex problem.

Once representation formats are defined, the protocols would also be extended to make the update packet more flexible. Instead of providing fields for only two articulated parts, a number of differentiated part fields would be allowed. The initial representation would provide keys into these fields to provide a correlation between the differentiated part field in the update and the change from a baseline representation. For example, a vehicle's thermal representation might be keyed so that a field in the update packet would indicate the amount of heat radiating from the engine compartment. In this example, the initial infrared representation description would correlate this field with a range of colors used to display the polygons comprising the engine compartment. The simulation host providing the representation could then use as few or as many of these fields as needed, based on the level of fidelity to which it was to be represented.

Similarly, remote event representations would be keyed to the network messages that indicate their occurrence. For example, when a weapon is fired, a message indicating the location of the muzzle blast is broadcast. This packet would key the aural and visual representations of the muzzle blast, based on the type of weapon fired.

Radar and sonar representations are more difficult to define, since these representations depend heavily on the capabilities of the emitting and receiving hardware. Conceivably, the sensor-equipped vehicle could generate the representation based on the polygonal data. However, this would require detailed knowledge about the materials from which the represented vehicle was composed. Because of the complexity of this interaction, further study will be required to define a consistent mechanism for distributing reflected representations.

Timing of Representation Distribution

Two approaches are possible for distributing initial representations in a timely manner that does not interfere with ongoing exercises.

Pre-Simulation Distribution. One possibility would be to have representations exchanged between network entities before an exercise begins. Typically, vehicle simulators are powered up and left in an "idle" state for a period of time before the exercise actually starts. This time could be used to pass representations among network entities.

However, using this initial time to exchange representations has several problems. First, according to the protocols currently used by SIMNET and the draft DIS standards, a vehicle can enter the exercise at any time. This frequently happens when aircraft simulators are used in large exercises, since they are not brought up until just before an air sortie is required. Second, there would be a management problem in ensuring that every vehicle was up long enough to learn every other vehicle's representations. The most serious concern with pre-simulation distribution, however, is that a simulation host that can simulate multiple entities (i.e., SAF) would have to transfer a representation for every possible vehicle that it might simulate, thus negating the economies associated with distributing representations of only those vehicles actually involved in the exercise.

Real-Time Demand-Driven Distribution. Instead of pre-simulation distribution, it would be advantageous for network entities to transmit vehicle representations on demand, in real time. When a simulation host receives an update from an unknown vehicle type, it would request a representation from the sending entity. The representations would be sent over a period of several seconds, thus avoiding any interruptions in the ongoing exercise. The representations would be broadcast so that any entity could receive and store them, not just the enquiring entity.

This approach has several advantages. It requires no pre-exercise management to ensure representation transmission, nor does it restrict the ability of simulated vehicles to enter the exercise at any time. Also, it takes advantage of the potential economies of representation, allowing network entities to maintain representations for only those vehicles actually in the exercise.

Additionally, since any vehicle's representation can be obtained on demand, a simulation host has the freedom to manage other representations in a manner optimal for its own hardware and software configuration. If appropriate, the entity might choose to keep only the representations for vehicles that it expects to interact with in the course of an exercise. A tank operating well inland might choose to ignore the representations of a ship far out to sea. If the tank drove to a coast, however, it could still request a set of representations from the ship at any time.

Also note that a simulation host is free to store any vehicle representations that it receives. The representations of typically encountered vehicles would probably be stored locally to lessen the number of network transfers. To ensure that the stored representations are indeed valid, the update messages could include a checksum against which the stored representations would be checked.

For example, as shown in Figure 5, Vehicle C (fixed-wing aircraft) is preparing to engage Vehicles A and B (armored personnel carriers). Vehicle D (ship) will already have a set of representations for C, and vice versa. First, C will likely require a set of representations from B, since an aircraft typically executes engagements at longer ranges than armor. Shortly thereafter, B will request a set of representations from C. Since A is adjacent to B, it will likely listen for C's representations, thus avoiding having to pose a query of its own shortly thereafter. Since A and B are unlikely to interact with D in this scenario, they will probably choose to ignore D's representations.

However, this approach still has a serious drawback. While a Local Area Network (LAN) should easily have the bandwidth to allow representation transfer without interfering with an on-going exercise, LHNs might not. This is significant because as the use of distributed simulation grows, the use of LHNs to tie simulation sites together will also increase.

Fortunately, this problem could easily be overcome by attaching a "representation server" to the simulation network. This server would interact only with network entities requesting vehicle representations. It would store the representations for every vehicle in an exercise, and would respond to local requests for representations instead of the requests being honored remotely. A set of representations for a given vehicle type would never be transferred over an LHN more than once, which should be quite tolerable for all but the slowest of links. When in use, a representation server would require the LHN gateway to screen out all representation requests and responses unless they are initiated from the server.

Alternatively, any network entity that could differentiate between local and remote information could serve as a representation server. For example, the LHN gateway might double as the representation server, which would simplify the process of screening out locally honored representation requests. In fact, this design is so flexible that any individual simulation host could choose to act as its own representation server by keeping the representations of every vehicle in the exercise, regardless of proximity. This would be desirable only in very small installations, however, as it negates some of the advantages that a distributed representation system offers.

Figure 6 shows two LANs connected by an LHN. LAN A, which is heavily populated with simulation hosts, has a representation server attached. This server handles all representation requests from vehicles on LAN A, and instructs the LHN gateway to screen out representation requests. For vehicles on LAN B, the representation server will only make requests when it does not already have a representation. On LAN B, there are fewer simulation hosts, so no representation server is present. Instead, the server functions are provided by the LHN gateway.

CONCLUSION

The current system of maintaining vehicle representations will become increasingly strained as the number of types of vehicle simulators increases. A distributed representation system will solve this problem, increasing the interoperability between different types of vehicle simulators made by different manufacturers. However, before this system can be realistically implemented, both significant investigation and design work in the area of representation formatting are needed. Furthermore, update packets that allow flexible representation of differentiated parts must be defined. A prototype system would determine optimal representation transmission rates and facilitate the construction of a representation server.

Figure 1. The SIMNET architecture allows for a variety of manned vehicle simulators and person-directed computer controlled vehicles (SAF). A vehicle's image is independent of whether it is from a manned simulator or a SAF simulator.

Figure 2. The visual representation of simulated vehicles includes different model types, including burning, smoking, and catastrophically killed.

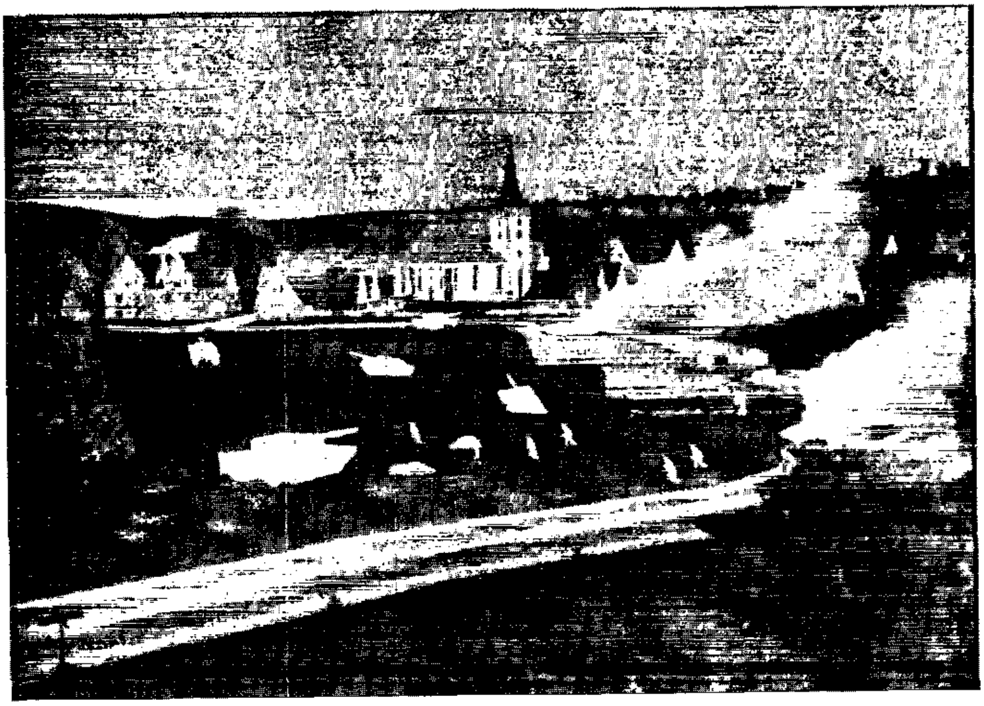


Figure 3. The infrared representation of simulated vehicles emphasizes the thermal signature of the participants.

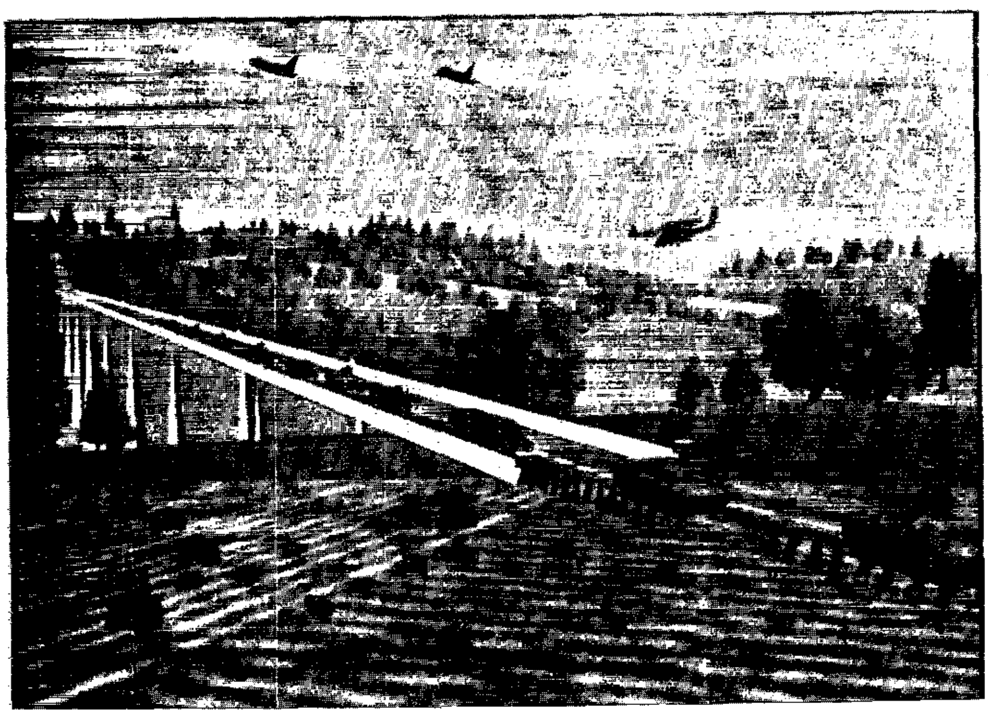
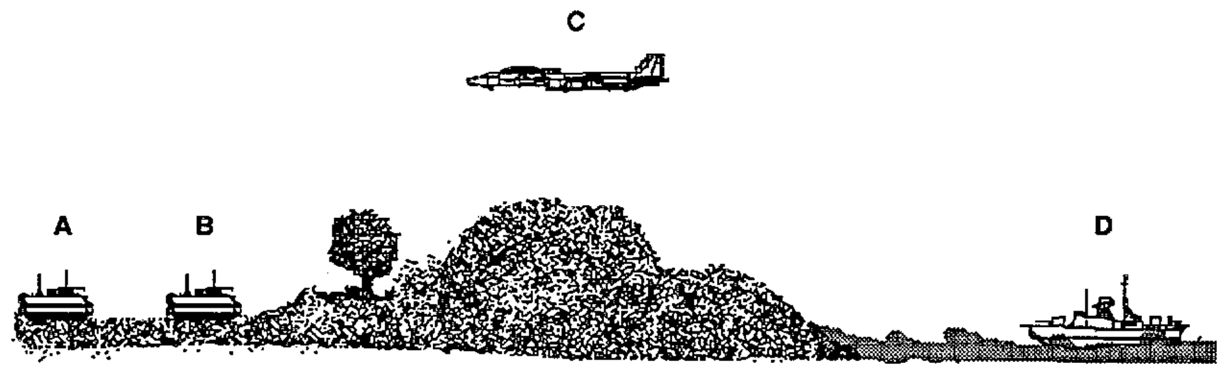
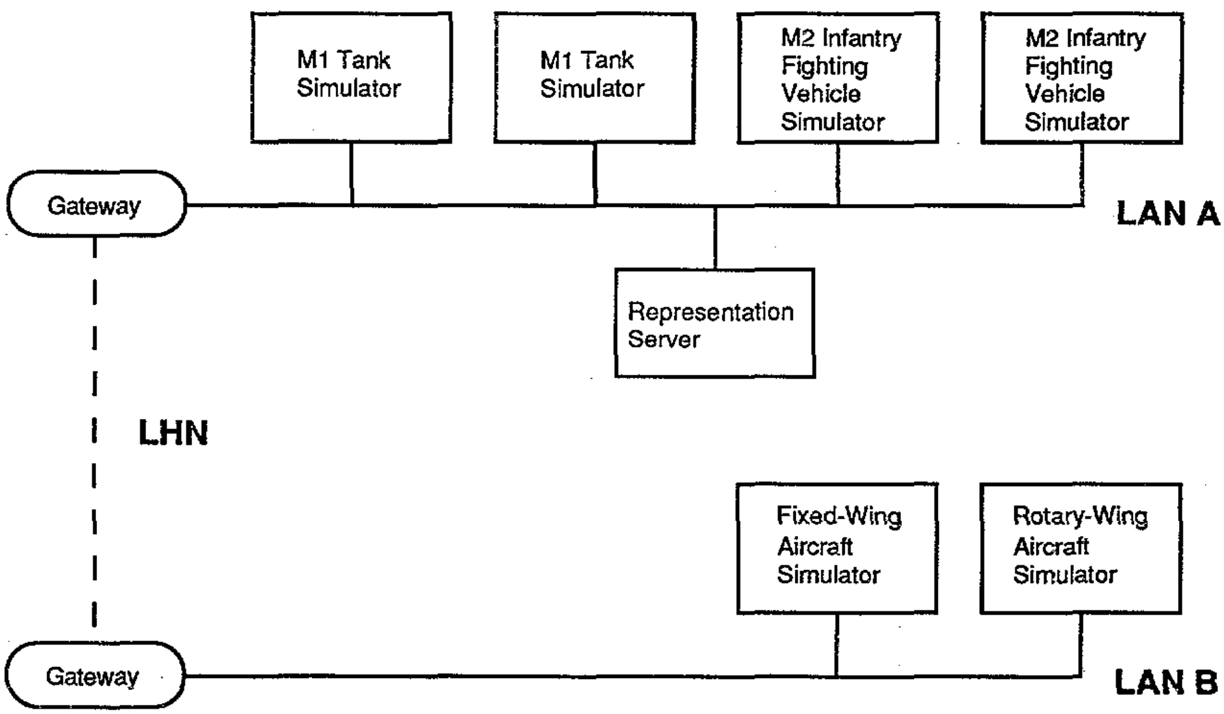


Figure 4. Large-scale exercises include many different types of simulated vehicles.



<u>Figure 5.</u> **Example of real-time demand-driven representation distribution: as Vehicle C flies across the terrain, it will encounter other vehicles and need their representations in real time.**



<u>Figure 6.</u> **Use of a representation server to facilitate representation distribution: LAN A has enough simulators to warrant a separate representation server, while LAN B does not.**

REFERENCES

1. Institute for Simulation and Training, "Military Standard (Final Draft) Protocol Data Units for Entity Information and Entity Interaction in a Distributed Interactive Simulation", 30 October 1991. For more information on obtaining a copy of this document, contact the University of Central Florida, Institute for Simulation and Training, 12424 Research Parkway, Suite 300, Orlando, FL 32826.

2. Pope, Arthur R. and Schaffer, R.L. "The SIMNET Network and Protocols", BBN Report No. 7627, June 1991. For more information on obtaining a copy of this document, contact the Defense Logistic Agency, Defense Technical Information Center (DTIC), Cameron Station Building 5, Alexandria, VA 22304.

3. Miller, Duncan C. "The SIMNET Architecture for Distributed Interactive Simulation", presented at the Summer Simulation Conference, Baltimore, MD, July 1991. For more information on obtaining a copy of this document, send a written request to Contracts/Documentation Requests, BBN Systems and Technologies, Advanced Simulation, 10 Moulton Street, Cambridge, MA 02138.

4. Project 2851, "(Draft) Military Standard Standard Simulator Data Base (SSDB) Interchange Format (SIF) for High Detail Input/Output (SIF/HDI) and Distributed Processing (SIF/DP)", April 1992. For more information on obtaining a copy of this document, contact PRC Inc., 1500 Planning Research Drive, McLean, VA 22102.

5. Gonzalez, James J. "Electromagnetic Propagation Modeling for Distributed Simulation", Proceedings: Thirteenth Interservice/Industry Training Systems Conference, Orlando, FL, 1991. For more information on obtaining a copy of this document, contact the American Defense Preparedness Association, Arlington VA (703) 522-1820.