A PROTOCOL CONVERTER

FOR

NETWORKED AIR DEFENSE APPLICATIONS

Daniel A. Bradford • Danielle M. Eriksen • Alan M. Thibodeau
Lockheed Sanders, Inc.
Nashua, New Hampshire

Huat K. Ng
Institute for Simulation and Training
Orlando, Florida

ABSTRACT

The standardization of training and simulation protocols through the SIMNET/DIS concept creates a unique opportunity to maximize the productivity of existing fielded systems. The adaptation of trainer protocols to SIMNET/DIS protocols will allow trainers to be integrated with distributed simulation networks, thus expanding SIMNET/DIS to better simulate and model the complete air-land battlefield. The focus of our work has been in the design and development of a general form protocol converter to adapt air defense trainers/simulators to a distributed simulation network. The air defense protocol converter incorporates generic air defense processing models, a menu driven user interface for customizing the protocol converter, and an object oriented design with clearly defined interfaces between models. The protocol converter houses all of the SIMNET processing needed for air defense systems to participate in a SIMNET exercise. In addition, we have defined a minimal set of messages between the air defense trainer and the protocol converter, which contains typical information available to the trainer. The protocol converter uses information from the generic interface message set and the user interface files to provide an air defense application with an intelligent interface to SIMNET, which can be easily expanded to address DIS.

ABOUT THE AUTHORS

Daniel A. Bradford received his Bachelor's degree in Computer Engineering Technology from Northeastern University in 1989. He has been involved in the planning and execution of various IR&D programs to expand the interoperability of training systems at Lockheed Sanders. He is presently the Principle Investigator of the SIMNET/PATRIOT Protocol Converter Research and Development Program upon which this paper is based. Mr. Bradford is a member of the Tau Alpha Pi Honor Society.

Danielle M. Eriksen is a Senior Software Engineer in Lockheed Sanders' Information Systems Division. Ms. Eriksen received her Bachelor of Science degree in Computer Science from the University of Lowell in 1985. She has been involved in software development on several major programs within Sanders, including a real-time simulation of an airborne electronic countermeasures system.

Huat K. Ng has received a Bachelor of Science degree in Computer Engineering and an Master of Science degree in Electrical Engineering from the University of Central Florida in 1986 and 1989 respectively. He has been working with the Institute for Simulation and Training since 1990, and his work is primarily focused on real-time simulation networking. He has been involved with Distributed Interactive Simulation (DIS) Standard development and implementation of application level gateways to interface dissimilar simulation protocols.

Alan M. Thibodeau is a Senior Principal Software Engineer in Lockheed Sanders' Information Systems Division. He received his Bachelor of Science Degree in Physics from Lowell Technological Institute in 1974 and his Master's Degree in Computer Engineering from the University of Lowell in 1979. He has been involved on several large trainer system projects, most recently on the PATRIOT Operator Tactics Trainer (OTT). He is presently assigned as the Program Engineering Manager for Air Defense Trainer related projects.

A PROTOCOL CONVERTER

FOR

NETWORKED AIR DEFENSE APPLICATIONS

Daniel A. Bradford • Danielle M. Eriksen • Alan M. Thibodeau
Lockheed Sanders, Inc.
Nashua, New Hampshire

Huat K. Ng
Institute for Simulation and Training
Orlando, Florida

INTRODUCTION

Advances in simulation technology coupled with a decreasing defense budget have created an environment which warrants increased simulation modeling for battlefield environments and an increased desire to upgrade existing fielded systems. Work initiated under the DARPA SIMNET (SIMulation NETwork) program has demonstrated the ability to develop simulations, force on force, that can create a realistic electronic battlefield environment. The Battlefield Distributed Simulation - Developmental (BDS-D) program of STRICOM has begun to address extensions that will create the entire "electronic battlefield." With the definition of the Distributed Interactive Simulation (DIS) protocols reaching a version 1.0 milestone, the remaining tasks will disseminate the architecture to platforms, systems and technologies that will support the entire electronic battlefield.

The use of SIMNET-based protocols has resulted in a significant new approach to training both large and small combat units. SIMNET is currently operational at training sites in the United States and Europe. Simulators are linked by a common real-time data communications network. Crews of these simulators can see and interact with each other against accurately scaled opponents on the same realistic battlefield to provide effective force-on-force training in a combined arms battlefield environment. The standardization of training or simulation protocols based on the SIMNET and DIS concept creates an opportunity to maximize the effectiveness of fielded training systems to provide large-scale training exercises without incurring the costs of transporting large numbers of personnel and equipment. Modification of existing fielded training systems to implement appropriate SIMNET protocols is a cost-effective way to upgrade existing assets to provide new functional capabilities while retaining currently implemented training functions such as instructor monitoring, student assessment and grading.

The adaptation of trainer-specific protocols to SIMNET protocols will allow existing trainers to be integrated with SIMNET networks, thus expanding SIMNET to better simulate and model the complete battlefield. Additionally, the quality of the local training system is increased as SIMNET engagements provide a more realistic range of command and control as well as combat service support elements found in actual military operations.

While many existing training and simulation systems are based on incompatible hardware and software architectures, suitably capable trainers can now be adapted to interact in the SIMNET environment through the use of a protocol conversion process. This process can be as basic as simply translating relevant trainer-specific information to and from SIMNET protocols, or can be made more intelligent to house additional SIMNET/DIS processing which the trainer/simulator is not concerned with.

BACKGROUND

The SIMNET/PATRIOT Protocol Converter (SPPC) project, funded by STRICOM, has researched the feasibility of linking a fielded air defense trainer/simulator to a distributed simulation network through the use of an intelligent protocol converter. The research, design and implementation of the protocol converter are the focus of the program. The air defense trainer (ADT) used in this research is the PATRIOT Operator Tactics Trainer (OTT) originally developed by Lockheed Sanders, Inc. for the US Army. The trainer has gone through recent modifications to incorporate technology upgrades and is presently fielded in Japan. As part of this upgrade, a "virtual console" was created which is a workstation based hardware simulation of the actual console. It is this virtual console and the Japanese version of software which have been used as the original baseline system in this research. The simulation networks researched are SIMNET (SIMulation NETwork) as described in "The SIMNET Network and Protocols Report No. 7627", and DIS (Distributed Interactive Simulation) as described in the document entitled "Military Standard (Final Draft) Protocol Data Units for Entity Information and Entity Interaction in a Distributed Interactive Simulation, publication IST-PD-91-1".

The current Lockheed Sanders PATRIOT OTT Fire Platoon (FP) was examined to determine whether it was feasible to connect it to SIMNET. The PATRIOT OTT FP software did not have the capabilities required to communicate directly to SIMNET or a Protocol Converter. It was configured to communicate with an instructor's console, and to simulate air defense using preprocessed target data. Under internal Lockheed Sanders funds, modifications were made to allow communications with the protocol converter and to allow for the real time operation of the FP software.

Detailed analysis of the SIMNET PDUs identified which fields in each message could be supplied directly by air defense trainers, and which fields from incoming messages would be needed by air defense trainers. Because air defense trainers will potentially vary widely in their level of fidelity and their capabilities, we attempted to keep the number of parameters to a minimum. This has maximized the intelligence and processing required by the Protocol Converter itself.

The SIMNET Protocols have been designed mainly for visual simulations; therefore, the message sets do not have complete data that would normally be required for sensor-based detection of airborne threats. The Protocol Converter development needed to compensate for this lack of information, and calculate default data for both incoming and outgoing PDUs.

After identifying those fields needed by the Protocol Converter to correctly process SIMNET PDUs, a generic interface was designed for communication from the Protocol Converter to air defense trainers. Use of the Protocol Converter to participate on SIMNET now requires that a trainer is modified only to implement this interface, which is far less complicated than the complete set of SIMNET protocols.

APPROACH

Our design has addressed two primary problems related to distributed simulation. First, how to modify existing, fielded training systems to be SIMNET compatible. Second, how to overcome the wide area network (WAN) implementation issues, such as bandwidth limitations and network delay. The design has been driven by our desire to minimize the changes to the existing trainer software and hardware, isolate the trainer local area network (LAN) traffic from the SIMNET LAN, and to develop modular/generic software.

To minimize the changes to the existing software and hardware, our first approach was to listen to the trainer network for SIMNET applicable information, read it off the network, and build PDUs to be sent to SIMNET. This approach made use of existing trainer messages where possible, with modifications to existing messages where necessary. New messages would only be created when the information was not on the trainer network at the required time.

This approach dictated that we have a working knowledge of the internal trainer software operation to allow us to create the appropriate SIMNET PDUs. It also assumed that the majority of the necessary information was present on the network at one time or another. If the information was not available, considerable changes to the trainer might have been required, which violated a key design goal. If a single event on the trainer LAN did not correspond to the issuance of a SIMNET PDU, history might have had to have been kept at the Protocol Converter (PC) to determine when a PDU should be sent. This approach was not pursued due to the constraints that it placed on both the trainer manufacturer and the protocol converter.

The present design is based on the protocol converter being placed on the SIMNET LAN, with a low speed WAN connection between it and the air defense trainer (see Figure 1). The protocol converter contains several models that perform air defense related processing. The design goal of minimizing changes to existing software has been met by defining a generic interface between the ADT and the PC. This allows the trainer manufacturer to implement the interface in the most efficient, cost effective manner possible. It also eliminates the requirement of having an understanding of the trainer software, and of the trainer having SIMNET relevant information on its network at specified times.

The design of the protocol converter also addresses the problem of WAN bandwidth and network delay for distributed simulation. By locating the protocol converter on the SIMNET LAN and connecting it to the trainer via a low speed WAN, we have been able to reduce the network bandwidth requirement and associated delays. This is possible because the time-critical processing related to air defense operations has been moved from the trainer location to the SIMNET location. These processing models have been designed to be generic and customizable to allow the maximum flexibility for use with air defense trainers/simulators. The co-location of the PC to SIMNET and its gateway design provide separation of the trainer LAN traffic from that of SIMNET, thus reducing the potential added traffic on the SIMNET LAN.

PROTOCOL CONVERTER DESCRIPTION

The Protocol Converter consists of six major software components; a Man-Machine Interface (MMI) - which is a separate executable from the actual Protocol Converter, a Protocol Translator, a SIMNET Vehicle Model, an ADT Vehicle Model, a Detection Model, and a Missile Model. All of the models have been implemented in Ada on a SPARCStation running Unix. The Ethernet communications portion of the PC was implemented in 'C'. The MMI was also implemented in 'C', using the X Window System for the actual user interface.

The air defense trainers that we investigated are concerned only with operator training for target identification, acquisition and engagement. They do not provide training for Logistics functions such as resupply and repair. The current version of the protocol converter does not model the Logistics functions as they are described in SIMNET.

To accommodate different air defense trainers, and also to allow individual air defense trainers to test different tactical strategies, there are several portions of the Protocol Converter which are user-configurable upon program start-up. The parameters are configurable through a menu driven interface which we have called the MMI (see figure 2), and are entered to the Protocol Converter in a format consistent with SIMNET PDUs whenever possible. For example, a vehicle's default appearance will be described in terms of Guises, Appearance, Markings and Capabilities. Parameters which describe how the detection model and missile model will function, such as subvolume ranges and missile fuse distance, are

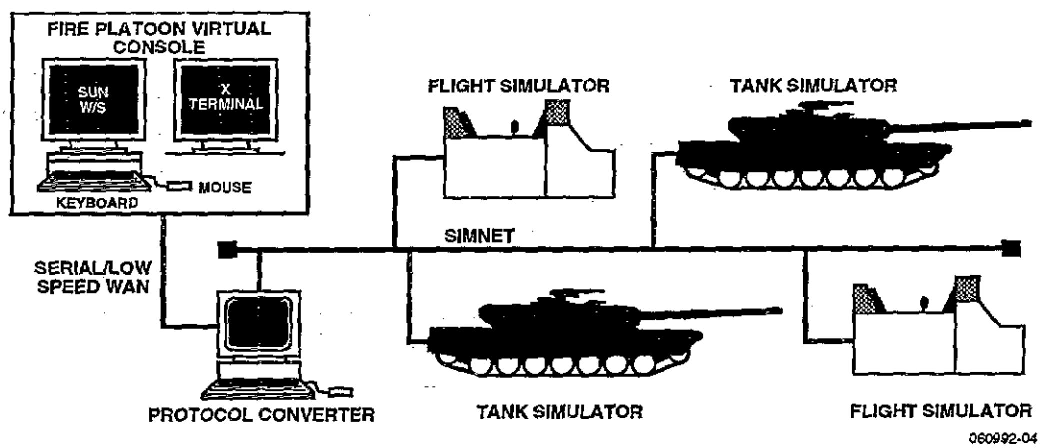

Figure 1. System Configuration

also configurable, even though they might not be used for PDU generation. These configuration parameters may still not allow all air defense trainers to use this protocol converter. They should, however, allow the flexibility needed for trainers and simulators of radar-based systems to easily use the system.

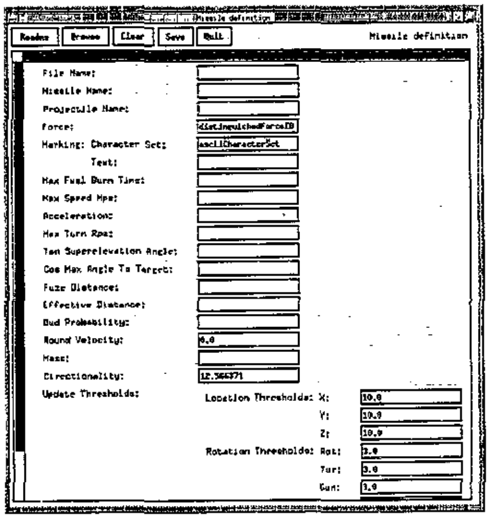

Figure 2. MMI Sample Menu

The first component of the Protocol Converter itself is the Protocol Translator, which provides a bi-directional translation between SIMNET format and the Protocol Converter's internal format. All messages received from SIMNET are examined and either sent to the appropriate function of the Protocol Converter or discarded. Messages from the Protocol Converter are translated into the proper SIMNET format and padded with the appropriate headers and any additional default data from the MMI that the ADT could not supply. A change to a different protocol, such as DIS, will have the most effect on this component.

The remaining components of the PC are divided according to the objects that they represent in the simulation world, that is, ADT Vehicles, SIMNET Vehicles, missiles, and radars (or other devices which perform detection). The ADT Vehicle Model and the SIMNET Vehicle model are mainly responsible for maintaining data regarding ADT and SIMNET vehicles and processing them according to the SIMNET protocols. The missile model and the detection model use information from the ADT and SIMNET Vehicle Models for their respective processing (see Figure 3).

The SIMNET Vehicle Model maintains the current information for all of the SIMNET vehicles of concern to the ADT. These are normally only airborne threats, but may also be land vehicles. It receives Vehicle Appearance PDUs from SIMNET, and first performs some basic Field of View filtering to determine whether the vehicles should be maintained or not. Potentially detectable vehicles are maintained in a list according to SIMNET protocol requirements - that is, their positions are dead reckoned until the next update is received, and they are removed from the list if updates are not received within the required time frames. This list is traversable by the other models of the PC that need information about vehicles playing on SIMNET. If visual updates are required by the trainer, vehicle update messages will be sent from the SIMNET Vehicle Model. These vehicle update messages would also be used in the absence of the Detection Model (detection processing being performed at the trainer), or if the trainer used both visual and non-visual detection methods.

The ADT Vehicle Model maintains a list of vehicles that are part of the air defense trainer, such as radars and launchers. Vehicle Appearance PDUs are sent to SIMNET for each vehicle in the list according to the SIMNET rules for static vehicles. Damage assessment is performed upon receipt of Impact, Indirect Fire, and Collision PDUs. Control PDUs from SIMNET, such as Activate and Deactivate messages, are also processed and responded to. The ADT is informed if a vehicle is destroyed or deactivated by SIMNET.

The Detection Model is currently implemented as a generic radar model, although the package specifications were designed to hide this fact as much as possible, and the bodies could therefore be implemented to simulate another form of detection. The model is capable of handling multiple radars, each with multiple subvolumes for search and track processing. The detection model scans the vehicles maintained by the SIMNET vehicle model, performing a series of tests on the vehicles to determine whether each is detectable. If detectable, the vehicle becomes a track and an update is sent to the trainer. The Detection Model sends Radiate PDUs to SIMNET as required by the protocol.

The Missile Model computes the missile position as it chases the target, and sends update messages to SIMNET and the trainer on current positions. The Missile Model receives engagement messages from the trainer for an unlimited number of targets. A target can be either a track from the Detection Model or a target that was engaged visually and maintained by the SIMNET Vehicle Model. The model also handles outgoing SIMNET PDUs, including Fire, Vehicle Appearance, and Impact PDUs.

The level of fidelity for simulating the Missile Model and the Radar Model were major issues of concern for the SPPC project. The simulation had to be realistic enough to mimic the tactical system in a training environment, yet fast enough to perform on a single processor within the real-time constraints of a SIMNET exercise. These models will be discussed in greater detail below.

GENERIC RADAR MODEL

An investigation of radar modeling was undertaken to determine what level of fidelity was appropriate for the Detection Model of the SPPC, and whether any suitable software already exists. This investigation mainly targeted the Lockheed Sanders' Technical Library and Modeling Labs, and revealed that the majority of the radar models are engineering models that typically run in batch mode and generate a very high fidelity output. While the detailed modeling can theoretically provide small computational errors under common circumstances, it is felt that some of the computational corrections to the general radar equation are not needed for training. Documented algorithms of generic radar simulation and the equations provided in basic radar textbooks provided the fundamentals required for specifying a less accurate model that could be executed in real-time, even under heavy target loads.

Using the knowledge gained from this investigation, we designed a generic radar model which utilizes a multi stage process to determine the existence of tracks. One process ascertains if a target falls within the appropriate search or track coverage area, and also which targets are illuminated. Another process determines the intervisibility between the ADT radars and targets. The final process determines if sufficient target signal exists for track identification, while accounting for some signal losses. This processing should be done at the Radar Scan Rate for the particular radar.

A primary objective of the Generic Radar Model for the SPPC was to be able to handle an unspecified number of radars, as well as vehicles and missiles, in real time. Principally due to this reason, the algorithms of the Generic Radar Model do not account for many potential factors affecting the received radar signal, such as clutter. Environmental effects that have been left out are air temperature, pressure, humidity, the curvature of the actual radar beam due to refractive effects, and atmospheric losses due to beam scattering. Other effects ignored are radar specific and account for the physical

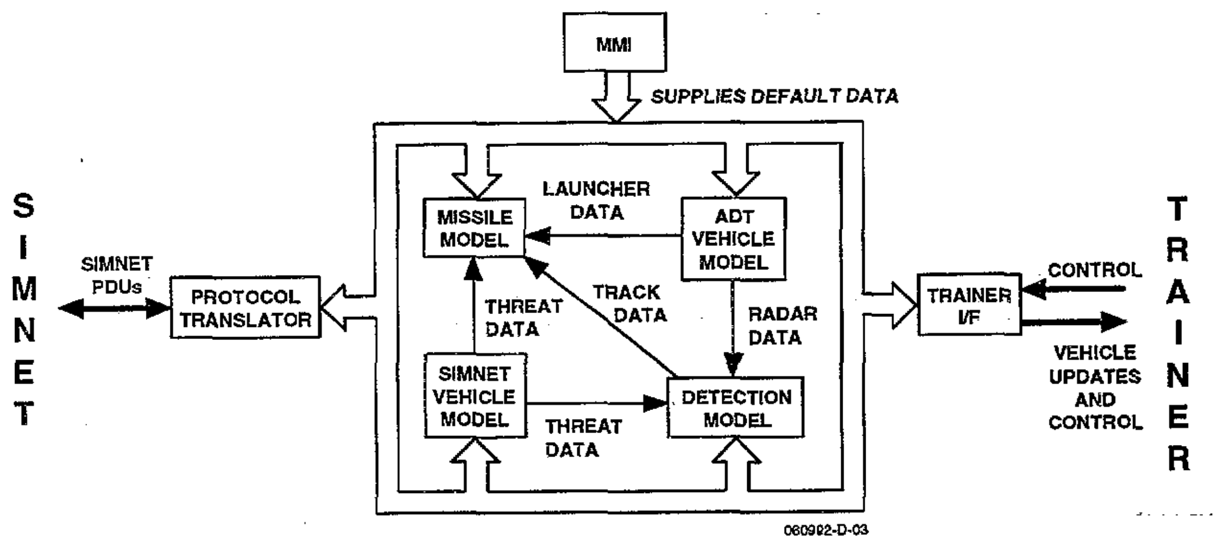

Figure 3. Protocol Converter Component Relationships

characteristics of the transmitter and receiver antennas, and losses of the receiving equipment. Finally, the Generic Radar Model is based on a simplistic mono-pulse radar; it does not provide for a Moving Target Indicator discernible by a Doppler radar.

The Generic Radar Model is parameter driven in order to represent many types of air defense radars. These parameters are provided to the generic radar model using two different methods. The first method provides parameters through the use of the MMI, as mentioned above. These defaults may be superseded by the second method of parameter specification, via messages from the ADT. This method allows for the ADT to dynamically create and modify simulated radars. This method should be utilized if possible, since a radar's parameters will normally change during the execution of a scenario.

Stages of Processing

The Radar Processing consists of a Basic Angular Coverage Check, Terrain Masking, Specific Coverage Area Check for Track and Search, and Signal Strength Evaluation (see Figure 4). The Basic Angular Coverage Check is the first test for each vehicle, and determines if it falls within the minimum and maximum azimuth and elevation of the combined search and track coverage areas. As search and track subvolumes (explained in a following section) are added, modified, or deleted, the overall minimum and maximum azimuth and elevation for these areas are maintained. This overall coverage area is used as a coarse filter to quickly eliminate vehicles that are neither detectable nor capable of directly jamming the radar. Vehicles that fall outside the overall coverage area are excluded from further processing.

Next, the Terrain Masking Check tests to determine if a vehicle's detection or jamming signal is blocked from reception by terrain. To perform this processing, a Terrain Masking Database is generated whenever the position of the radar is specified, using the terrain elevation data extracted from a SIMNET Database Interchange Specification (SDIS) database. The Terrain Masking Database contains a list of the sine of the elevation peaks for each radial, of four degree azimuth width, centered at the radar location. The sine of the elevation angle of the vehicle and its ground range are then checked against the appropriate radial from the Terrain Masking Database to determine if the vehicle is blocked by a closer terrain peak. If the vehicle is masked by terrain, no further processing is performed on it.

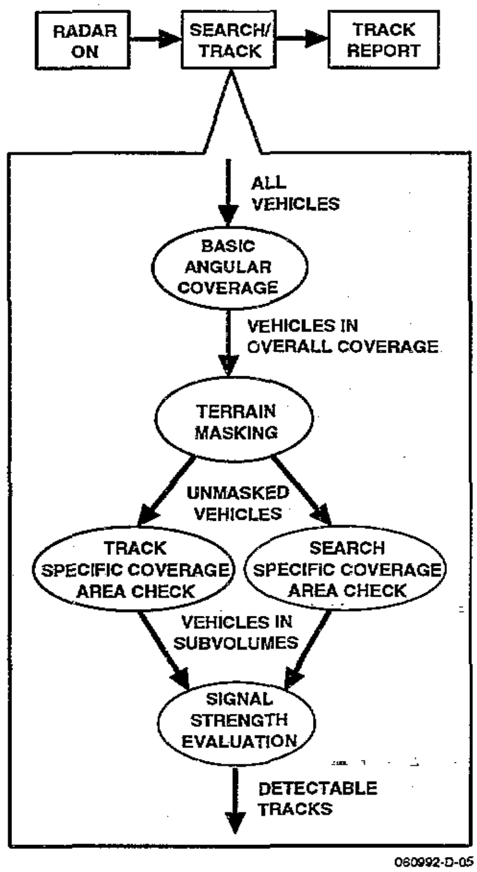

Figure 4. Radar Model Processing Diagram

The total noise signal considered by the Generic Radar Model which interferes in the detection of a vehicle is composed of the radar receiver noise signal plus the signal due to jamming for a particular spherical sector plus the returned chaff signal. Since electronic countermeasures (ECM) and chaff are not included in the SIMNET specification, the current implementation only models radar receiver noise. Jamming and chaff are planned for inclusion in the DIS specification.

The Specific Coverage Area Check identifies whether a vehicle is within a search or track subvolume. A subvolume is a spherical sector described by minimum and maximum Range, Azimuth, and Elevation Angles (see Figure 5). The Azimuth Angles are specified relative to the boresight, or Primary Target Line (PTL) Azimuth Angle. The

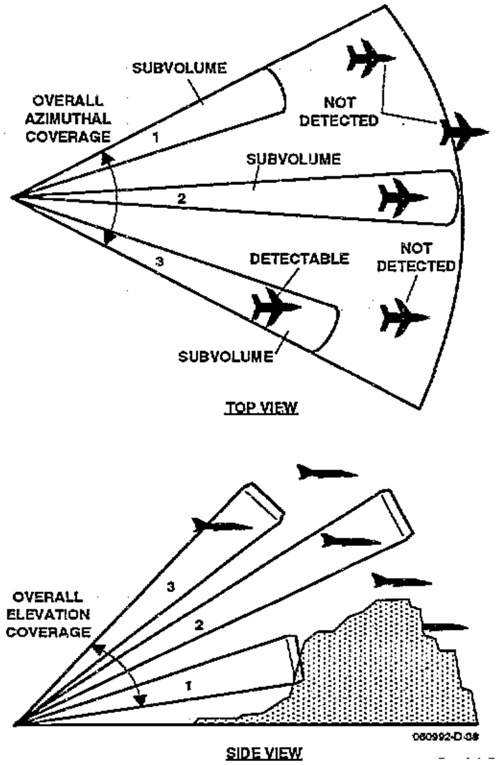

Figure 5. Specific Coverage Area

Elevation Angles are specified relative to the PTL Elevation Angle. To generalize the search and track volume specifications for a radar, an unlimited number of subvolumes describing the volume is allowed. Each subvolume will have a Subvolume Reference ID so that individual subvolumes can be dynamically created, modified, and deleted.

For each radar, dynamic lists of the current track and search subvolumes are maintained. Each previously tracked vehicle is checked against the appropriate track subvolumes to determine if it is still within tracking space. If not, drop track processing is performed on the track. Untracked vehicles are checked against the appropriate search subvolumes to determine if they could be detected in this cycle.

One aspect of this processing is the determination of which vehicles are illuminated by the radar during this scan. All vehicles that are subjected to the coverage area check and that fall within the spherical sector for a search subvolume, ignoring the range limitations, are flagged as being illuminated. This data is required later for the generation of the Radiate PDU.

All vehicles that pass the Specific Coverage Area Check are subjected to Signal Strength Evaluation. A basic radar equation is utilized for computing the returned radar signal from a vehicle. The first step is to compute the Target Power Signal:

$$Target\_Power\_Signal = (Power * Gain^2 * Attenuation^2 * Speed\_of\_Light^2 * Radar\_Cross\_Section) / (4\pi^3 * Target\_Slant\_Range^4 * Frequency^2)$$

The Generic Radar Model requires a Radar Cross Section for a particular vehicle in determining the returned signal strength. Actual radar cross sections of vehicles depend on frequency and vary with small incidence angle changes. Accurate modeling of radar cross section requires huge databases indexed by radar, vehicle, and frequency for each roll, pitch, and yaw incidence angle. Such a database was considered beyond the scope of this project. Instead, a coarse approximation of a vehicle's radar cross section, allowing for some incidence angle variation, was developed. The radar cross section is obtained from a Vehicle Data File which contains records for individual vehicle types.

Each vehicle entry contains an approximate radar cross section for several Z-axis angles. The Z-axis of a vehicle is perpendicular to the plane of its top or bottom. Although the X-axis and Y-axis aspect angles also affect the radar cross section, the coarse approximation only takes into account the radar to Z-axis aspect angle at 30 degree increments.

Next, the Target Power Signal is compared to the radar receiver noise. If the Target Power Signal is greater than the noise then the vehicle is flagged as detected.

One or more Radiate PDUs must be generated after completion of this scan. All vehicles that have been "detected" are included first in one or more Radiate PDUs. Then, all vehicles that were illuminated but not detected are listed.

GENERIC MISSILE MODEL

As with the Radar Investigation, an investigation was undertaken to determine the availability of a suitable generic missile model, either in code or algorithm. At the outset, it was established that real-time modeling of multiple missiles against a potentially large number of targets was critical. Missile algorithms that modeled the detailed fin action or provided 6 degrees of freedom (6-DOF) were considered inappropriate because they are both computationally intensive and beyond the fidelity required by the visual simulations of SIMNET/DIS. The investigation encompassed both a library search and an internal review of available missile models documented in Sanders' Modeling Lab.

Two "generic" missile models were reviewed which included code for the entire engagement problem. This included radar detection, launcher selection and motion, aerodynamics, guidance, fuzing, seeker, system response times and reliability, trajectory, ECM, and missile vulnerability. These models could not be used for two principle reasons - first, the modeling of the processes was more detailed than desired, and second, much of the processing is performed by other parts of the Protocol Converter, or is not designed to be part of the PC at all. For example, the ADT is responsible for launcher selection and motion, system response times, and other system specific details. We also determined that modeling of missile target acquisition methods such as navigational tracking, passive/semi-active/active radar, home-on-jammer, track-via-missile, infrared, and visual is beyond the scope of the Generic Missile Model. The end result of the investigation was that no internal missile model was found that could directly be used for the SPPC. The existing SIMNET *6.6.1 Semi-Automated Force (SAFOR) Missile Model* source code was then investigated. This software is written in 'C' and is tightly coupled to the other SAFOR modules, which prohibited its direct reuse. Many of the algorithms, however, were used in the basic design of the SPPC Generic Missile Model design.

The characteristics of the modeled missile are input at the MMI; these include the maximum tracking angle, fuel, maximum turning rate, maximum speed, fusing distance, as well as the warhead and detonation types. The ADT is responsible for providing dynamic information to the protocol converter, such as the selected launcher and designated target. The ADT can also abort the missile at any time during the flight.

The Missile Model deals with the missile from launch until explosion or ground impact. The Missile Flyout portion of the Missile Model is designed as a state machine. At launch, missile flyout modeling begins and the missile is in the "Tracking" state. If it reaches Fusing Distance to the target, the missile is then "Armed". If the target falls outside of the Maximum Tracking Angle or if the vehicle has been dropped then the missile is "Lost". Finally, once the missile runs out of fuel, it begins a free fall and is "Crashing" (see Figure 6).

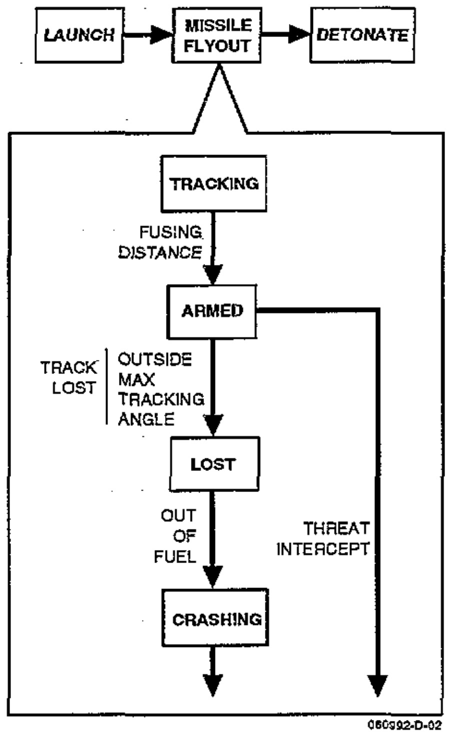

Figure 6. Missile Model State Transition Diagram

Missile Launch

The missile is launched as a result of an operator action in the trainer/simulator. The modeling of the missile is based on the kind of missile launched and corresponding characteristics previously input at the MMI.

Missile Flyout

The Missile Model flies the missile after the target using the basic laws of motion, ignoring details like friction and lift. The missile exits the tube in the direction of the launcher at its Initial Speed. The missile will accelerate at a constant rate up to a Maximum Speed. As long as the missile keeps the target within the Maximum Tracking Angle, it will attempt to match the tangential velocity of the target, to keep it in view. The remainder of the available speed is used for closing in on the target. There is no modeling of the target detection process except for the angular check.

Missile maneuvering is limited by a Maximum Turn Rate capability, measured between the original velocity vector and the updated velocity vector. If the missile loses track of the target, it will fly straight until its Fuel Time is expended. Once fuel is expended, the missile begins a free fall with its XY velocity components remaining the same and its Z continuously changing due to gravity, until impact with the ground (no detonation).

Missile Detonation

Once the missile is armed, it is checked for passing the point of closest approach, i.e. the range rate begins increasing. When this occurs, the missile is presumed detonated at the closest point. The missile may also be prematurely detonated by an ADT message.

The Missile Model is completely responsible for generating and issuing the required SIMNET messages. It will issue the Fire PDU, multiple Vehicle Appearance PDUs, Impact PDU, and the Deactivate PDU.

CONCLUSIONS

The SPPC program has confirmed that the protocol converter approach to linking an existing air defense trainer/simulator to a simulation network is a viable technical solution.

Using the Protocol Converter, the PATRIOT OTT has been successfully integrated with other vehicles playing on SIMNET. That is, a PATRIOT OTT operator has successfully engaged multiple tracks (aircraft SAFORs), maintained by the detection model, with multiple missiles flown by the Missile Model. As of the date of this paper, testing of the PC is still in process and the upper limits (number of tracks and targets) have not been determined.

The protocol converter has been designed to be configurable for use with different air defense trainers. Customization can be in the form of replacing entire generic models, such as the Detection Model, and by using the current set of configuration parameters provided through the MMI to modify the processing. Use of the protocol converter to participate on SIMNET now requires that a trainer is modified only to implement the generic interface, which is far less complicated than the complete set of SIMNET protocols. These elements have maximized the intelligence and processing required by the Protocol Converter itself.

By locating the protocol converter on the SIMNET LAN and connecting it to the trainer via a low speed WAN, we have been able to reduce the network bandwidth requirement and associated delays.

Upgrading the protocol converter from SIMNET to DIS version 1.0 will not require major modifications due to the similarities in the protocols. The differences between SIMNET and DIS have been investigated concurrently with the development of the PC, and the conversion task is presently in process.

LIST OF ACRONYMS

ADT	Air Defense Trainer
DARPA	Defense Advanced Research Projects Agency
DIS	Distributed Interactive Simulation
DOF	Degrees of Freedom
ECM	Electronic Countermeasures
FP	Fire Platoon
LAN	Local Area Network
MMI	Man Machine Interface
OTT	Operator Tactics Trainer
PC	Protocol Converter
PDU	Protocol Data Unit
PTL	Primary Target Line
SAFOR	Semi-Automated Forces
SDIS	SIMNET Database Interchange Specification
SIMNET	SIMulation NETwork
SPPC	SIMNET/PATRIOT Protocol Converter
STRICOM	Simulation, Training, and Instrumentation Command
WAN	Wide Area Network