A Distributed Virtual M&S Framework for Military Tactical Training

SeongKee Lee, ChanGon Yoo, JungChan Park, JaeHyun Park
Directorate of Information Software
Agency for Defense Development
Songpa P.O.Box 132, Songpa-Gu, Seoul 138-600
Republic of Korea
Sukyeongy@hanmail.net

Abstract: This paper shows a distributed simulation framework for tactical training in Networked Virtual Environment (NVE). These days the NVE technology for an operational tactical training system plays a significant role in military training courses. The existing military training simulation systems are mostly full simulated systems operating on single platform. They are to train individual's operation skill, but don't support team level tactical training. In order to train team level combat and command skill in dynamic battlefield, the interaction among distributed combat objects and diverse battlefield composition are required. This paper designs a distributed simulation framework to satisfy these requirements. The framework produces the networked virtual environment using virtual reality, event based simulation and HLA/RTI based interoperation techniques. This paper implements a small scale tactical training system based on the framework.

Key-Words: Modeling and simulation, Simulation Framework, Networked virtual environment, Tactical training

1 Introduction

Modeling and simulation (M&S) technology is an effective method to analyze alternatives, train operation procedure or develop optimal system. It models the real complex environments to the abstract level and simulates diverse solutions. As real environment becomes from static to dynamic and from single platform to system of systems, some requirements must additionally be considered as follows: 1) distribution. The objects are distributed. 2) interoperation/heterogeneity. The heterogeneous objects must be interoperable; 3) virtual environment. Since dynamic environment must be represented, it is effective to represent virtual environments; 4) autonomy. When scale becomes large, autonomous objects are required.

In order to satisfy the requirements or the trends of modeling and simulation technology described above, the networked virtual environment, NVE technique has been developed. Today, an operational tactical training system is considered indispensable in military area. This is mainly due to sudden rises in the price of petroleum, drastic cuts in Department of Defense spending, complaints of inhabitants living near exercise fields, and environmental constraints about the presence of endangered species of plants and animals. Reflecting the current and future situations, it is expected that demands for tactical training systems using NVE will rapidly increase since synthetic environments presented by virtual simulation systems are extremely cost effective. The NVE technique provides the synthetic environments so that distributed objects on the environments can interact with each other on virtual space.

In order to realize the networked virtual environment, some element techniques are needed: networking technique that distributes objects, simulation technique that supports the interaction among objects and virtual reality technique that virtually represents environment. In general, NVE systems for collective team training will involve large numbers of participants and simulators, so the systems should allow them to meet and interact cooperatively in real-time, which necessarily involves complex problems such as real-timeness, scalability, robustness, interoperability, and reusability. To solve that kind of problems this paper designs a distributed modeling and simulation framework based on the NVE technique, which is intended to assist simulation software developers in concentraining on simulation logic itself by providing a composable component-based programming environment when they construct scalable real-time simulations, and applies the framework to implement the battle simulation system for tactical training. The battle simulation would be one of typical systems that NVE technique is required. The system explained in this paper is a small scale tactical training simulation model that the distributed combat objects share with a virtual digitized 3D battlefield and engage each other under command. Each object can act within its capability: seek, move, fire, etc. In a word, the goals of the research project described in this paper are summarized as follows: 1) to identify common features of current tactical training systems operating in synthetic battlefields 2) to establish design rationale of the distributed real-time simulation system for military tactical training 3) to build a modeling and simulation framework to accurately and effectively represent combat models 4) Ultimately, to enable distributed real-time simulation developers to easily model simulation objects and make the objects work cooperatively.

Section 2 explains the general concepts to give a glance about modeling and simulation in military. Section 3 proposes distributed simulation framework based on NVE technique. In particular the design rational and architecture of the for tactical training simulation system are addressed. Section 4 implements tactical training simulation system based on the distributed simulation framework designed in Section 3. Section 5 discusses some technical issues to be considered on the framework.

2 Modeling and Simulation in Military

In military, they say that all but war is simulation. The simulation in military is essential tool to all areas of analysis, training and acquisition. In order to understand military simulation domain, this section explains some related concepts such as layers, type, purpose etc and classifies military simulation domain according to their combination.

2.1 M&S Layers

In general, military simulation can be layered as follows: 1) Engineering layer. This layer models and simulates to develop optimal system. Performance and effectiveness factors are tested. 2) Engagement layer. This layer is related to the system engagement capability. The 1 to 1, N to N engagement capabilities are tested in view of system. 3) Battle and mission layer. This layer simulates combat situation in real battlefield. Modeling natural and artificial battlefield environment, interaction among combat objects including command, display of environment, mission analysis etc are needed. 4) Theater layer. This layer analyzes theater level capability between friendly and opponent force using fire power index etc.

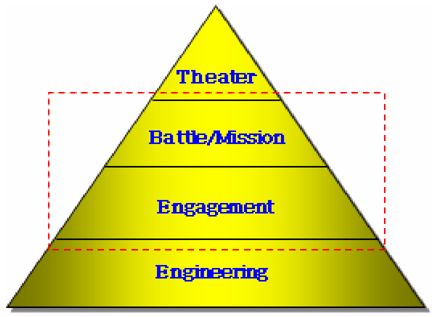


Fig. 1 M&S model

2.2 M&S Types

Military simulation can be also classified as Fig. 2: 1) Live simulation. This is simulation that doesn't fire real but use laser engagement system in real field. 2) Virtual simulation. This models real battlefield to virtual environment and simulates diverse situations on virtual space. 3) Constructive simulation. This represents combat unit by index such as fight power, fire capability etc. and simulates according to the predefined scenario.

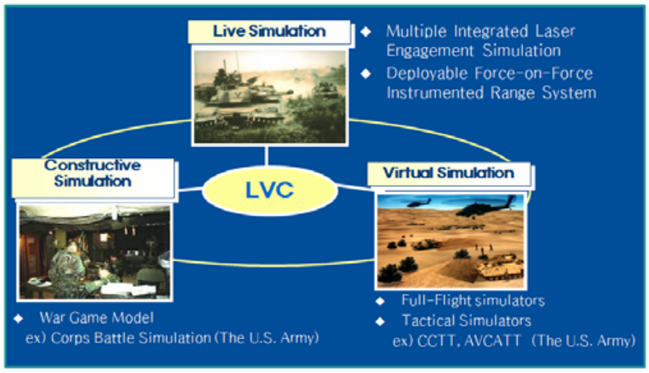


Fig. 2 L-V-C Simulation

2.3 M&S Purposes

As explains above, M&S tool in military is primarily used as follows: 1) Analysis. Command ability, fire power or fight capability are analyzed based on simulation. Also optimal COAs (course of actions) against opponent can be analyzed. 2) Train. Military doctrine and operation procedures can be educated and trained using simulation. 3) Acquisition. M&S is used to design system, measure performance and testing and evaluation.

2.4 M&S Scales

According to the application scale, military simulation can be classified as follows: 1) Small scale such as individual, squad and platoon level. 2) Middle scale such as company and battalion level.

3) Large scale such as brigade and division level.

2.5 M&S Forces

Basically, military simulation systems are used for 1) Army service, 2) Navy service, 3) Air force service.

Combining the views of Subsection 2.1 to 2.5, military modeling and simulation can be classified. For example, if simulation system is virtual simulation system for tactic training of army company level, then the system is classified into Engagement-Virtual-Train-Middle-Army, E2-V-T-M-A. Since 4 layers, 3 types, 3 purposes, 3 scales and 3 services are identified, the whole possible combinations are 324 ideally. If even a view is added, the combinations would be increased in exponential. Developing all combinations requires many efforts. That is not effective way. It may be an effective approach to develop combinations based on framework. For virtual simulations, networked virtual environment technique may be used to develop many combinations related to virtual simulation type. This paper considers networked virtual environment technique as a useful framework to develop virtual simulations.

3 Distributed Simulation Framework

In concept, framework is the architecture including technical view, operational view and system view that are required to construct system. Based on networked virtual environment technique, this section designs the framework for virtual simulation. This section discusses some considerations for virtual model and simulation framework.

3.1 Design Rationale

The framework must be designed to efficiently build and manage objects for simulation system development. In detail, the following principles must be considered in framework.

- Distributed Architecture: Typically objects in simulation are distributed. Central control for event scheduling and conflict resolution is not efficient. Each object maintains its own state and acts independently on system. The distributed environment may be local or wide. Network protocols must be supported.
- Interoperability: The framework aims to interoperation among heterogeneous objects on one battlefield environment. To support to this principle, interoperation method such as middleware must be supported.
- Real Time: The objects need to feel reaction in real time so that exercise participants can feel perceptual consistency in an NVE system. The multithread and synchronization methods can be used. Replication may be used to process large size data such as terrain data in real time.
- Composability: Since diverse environments are required, the framework must compose diverse situations. Also already developed objects or environments may be integrated into the framework. The units must be composed and operated together in run time. In result, model developers can build new models by combining existing one.
- Scalability: Simulation may be executed from small scale to large scale. Scalability is mainly accomplished by leveraging locality of references. The framework needs to support extendibility without much performance loss.
- Productivity: Tactical training application developers are able to easily build and manage simulation objects, and are able to concentrate on the design of simulation logic itself, not hampered by trivial programming problems.

3.2 Framework Architecture for Virtual Simulation

Reflecting the design principles discussed in Subsection 3.1 to networked virtual environment technique, the followings must be considered. To satisfy distributed architecture, network component is needed. For interoperation, simulation services such as middleware, message transfer and event processing and time management are needed. Also, in order to real time processing, system services such as multithread are required. In order to satisfy composability, every element composed of simulation environment, for example, entity, natural environment etc, is modeled into a model as a component using appropriate modeling technique. Also the simulation lifecycle modules necessary to prepare, control and execute simulation as well as the analysis of simulation result must be supported. Scalability is supported by hardware, network or interoperation service. To virtually represent simulation situation during and after exercise, 3D display module needs to be included. Also real data such as combat effect data and dynamic physics model must be constructed in database or library. These considerations can be included into a framework. The framework can be architected into layered style by mapping services or components to corresponding layers. Each layer plays its own role and is interconnected with each other. In short, the framework appropriate to virtual simulation system can be established as Fig 3. The framework may be general, but provides fundamental concept to construct simulation systems

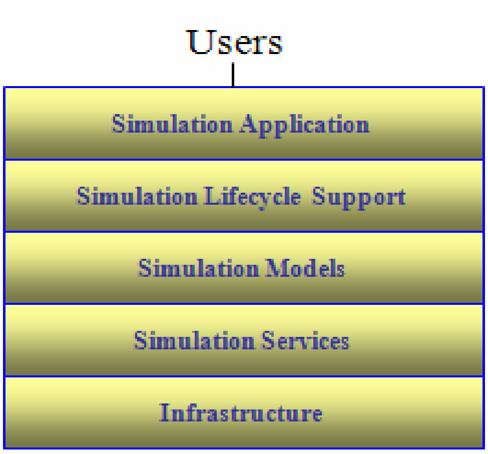


Fig. 3 Virtual M&S Framework (VMSF)

4 Tactical Training Simulation System (TTSS)

The Virtual M&S Framework (VMSF) in Fig. 3 must be used as a baseline to develop many virtual simulation systems. This section applies the framework shown in Fig. 3 to the development of a tactical training simulation system. The simplified system description is below.

This system is a battle simulation system which enables the participants to share with common 3D synthetic virtual battle-space and engage with each other on it. The exercise participants are commander, combatant, unit, weapon system, etc. All trainees are connected on local area network. The system provides the functions necessary to prepare, execute and analyze exercise. Also, in order to simulate diverse tactics, this system includes computer generated force with semi-autonomy. Since the system is a virtual simulation system for tactical training of army company level, it is classified as B-V-T-M-A combination, that is, battle layer-virtual type-training purpose-middle scale-army service and belongs to the dot box of Fig. 1.

4.1 System Design based on VMSF

Based on VMSF, this paper designs the tactical training simulation system described above. Here, design implies to specify the VMSF framework to the TTSS. Fig.4 shows the TTSS technical architecture (TTSS-TA) which is one of framework views: TA (technical architecture), OA (operational architecture), SA(system architecture). This shows the technical choices in layers of VMSF to construct TTSS.

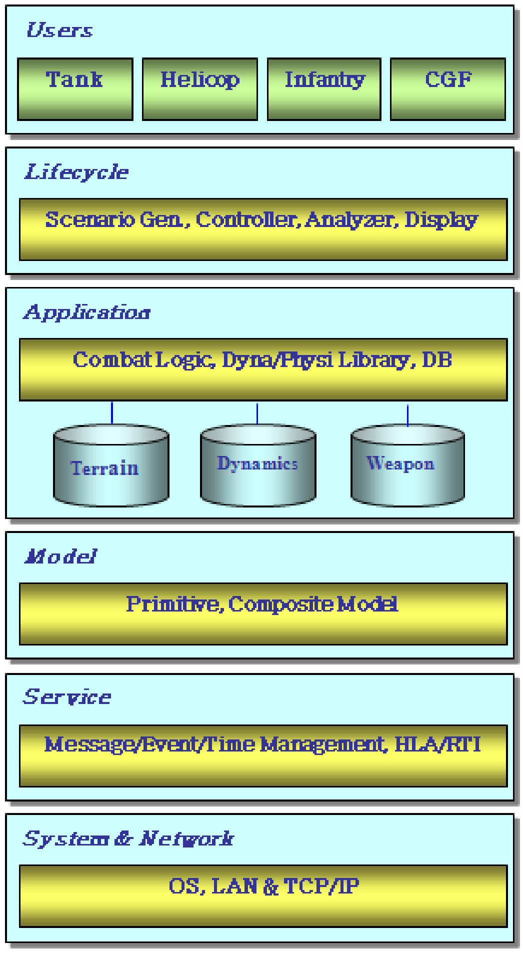


Fig. 4 TTSS-TA based on VMSF

Fig.4 is to map VMSF into TTSS-TA. Since infrastructure layer of Fig.3 includes components to run system, it is mapped into system and network in Fig.4 and includes network and operating system. Simulation service layer provides the services such as interoperation service, message distribution, time management and event management which are simulation engine functions. In particular, HLA/RTI software, which is a kind of middleware for defense simulation interoperation, is provided to support interoperation with TTSS users. Simulation model layer generates entity to be involved in simulation.

Models are composed of primitive model and composite model which is based on DEVS model. A primitive model is a basic model which has specification for the dynamics of entity. It describes the behavior of entity. A primitive model contains the following information:

- Set of *input ports* and *output ports*
- *Internal transition function* which is called when a registered scheduled event occurs
- *External transition function* which is invoked when an input event arrives at the model
- *Output function* which generates an external output
- *Continuous model function* which is called as often as possible by the simulation engine. The internal transition function and external transition function are mainly for discrete event simulation, while the continuous model function is for a continuous simulation operation.

A composite model shows how to couple several entity models together to form a new compound model. The subcomponent models may be primitive model or composite model. That is, a composite model can be employed as a component in a larger composite model, thus allowing the construction of complex models in a hierarchical fashin. A composite model contains the following information:

- *Input ports and output ports* through which external events are transmitted
- *External input* coupling which connects the input ports of the comsposite model to one or more of the input ports of the components
- *External Output* coupling which connects the output ports of components to the output ports of the composite model
- *Internal Coupling* which connects the output ports of components to the input ports of other components

Simulation lifecycle support layer in TTSS-TA provides the functions needed to prepare, execute, control simulation and analyze results: battlefield creator and display, scenario generator, exercise controller, after action review, etc. User layer provides virtual interfaces for real users. TTSS simulates weapon system such as tank and helicopter, infantry, CGF (Computer Generated Force) and commander. Real users participate at exercise through PC-window interface. In application layer, in order to show the effect of interaction(engagement) among entities, the libraries such as damage lib, dynamics lib, physics lib, rendering lib and some databases such as terrain data are supported.

The TTSS technical architecture of Fig.4 reflects some considerations described in Subsection 3.1. For distributed architecture, TTSS-TA provides TCP/IP protocol network and local area network environment. For interoperability, HLA/RTI(High Level Architecture/Run Time Infrastructure, IEEE 1516 standard), which supports interaction among diverse entities. For real time feature, TTSS-TA includes simulation engine which is executed by an event based simulation method. System services such as multithread are included. For composability, entity can be easily added and deleted by creating instance of class. The added entity can interact with existing entities together. Also, exercise participants can join and withdraw in run time. For scalability, TTSS-TA supports the interconnection between HLA/RTI and TCP/IP groups. Since HLA/RTI may reflect to system performance by the number of participants, TCP/IP network is used to expand participants. Fig.5 shows the interconnection concept between HLA/RTI and TCP/IP groups to expand simulation scale to company exercise at least.

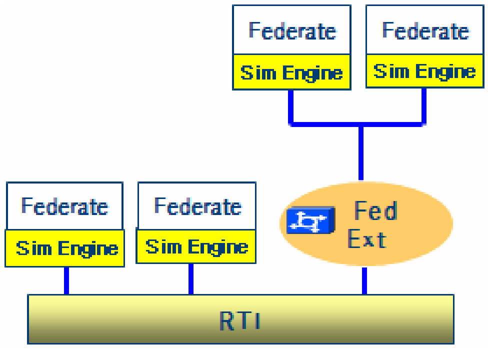


Fig. 5 Simulation Expansion

4.2 System Implementation

Based on the TTSS-TA of Fig.4, company level tactical training simulation system, TTSS is implemented. At first, using the components necessary to construct TTSS, this paper designs system architecture in detail. The system architecture is composed of basic modules and interface modules.

Basic modules are essential functions in simulation execution. All components except users interface are included as follows: battlefield environment creation, combat logic, scenario generation, exercise control, simulation engine, after action review and combat situation display.

Interface modules are interface for participants to join in exercise. TTSS includes interface for individual and weapon system as following: infantry, tank, helicopter, commander and Computer Generated Forces (CGF). Real users show virtual window interface and execute exercise by manipulating joystick or keyboard. The interactions among users are supported by RTI software. RTI plays a broker to interact with users. That is, each user interacts with others via RTI software. To expand exercise users, federation extender, which is light RTI and acts like a federate of RTI, is implemented as shown in Fig.6.

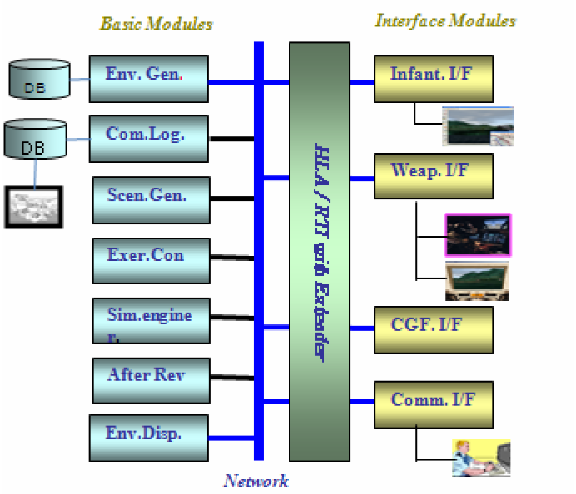


Fig. 6 TTSS Modules

Fig.6 is implemented on PC based environment in which Window XP is used. For interaction among users, MakRTI software and federate extender of Fig. 5 has been used. For message transfer, XML format has been used. Programming language is C++. For effective development, object oriented development methodology is used during whole development cycle. Most modules of Fig.6 have been coded in manual rather than COTS products. This approach is much effective to reuse and upgrade software modules.

The software modules support whole exercise cycle of pre-exercise, exercise and post-exercise as Fig 7.

- Pre exercise: battlefield authoring by environment generator module, combat effectiveness data preparation by combat logic module and exercise situation deploy by scenario generation module.
- Exercise: simulation execution by simulation engine module, simulation start/stop, battlefield situation change, logistics' supply etc by exercise controller module, exercise situation display by environment display module, exercise execution by infantry interface, weapon interface and commander interface.
- Post exercise: exercise result analysis by After Action Review (AAR).

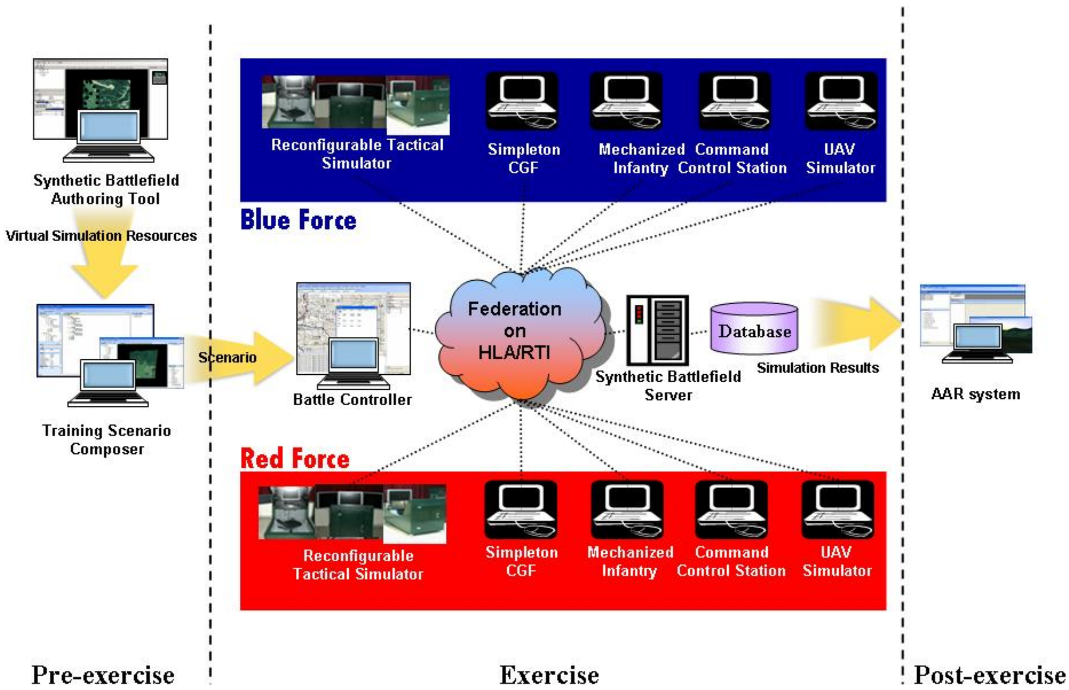


Fig. 7 TTSS Exercise Phase

Once a synthetic battlefield and a training scenario are prepared for an exercise, the Exercise Controller assigns PCs to exercise users. The assigned user logs in the corresponding PC, and according to own role such as unit, infantry, weapon or commander, participates at exercise through interface. Exercise controller starts and stops the whole simulation, changes battlefield environment such as rain, snow, fog, dawn, night and etc, supplies logistics by request and plays as a high level unit or commander such as indirect fire supports. Each users share battlefield and engage within their combat capability. Interaction among users is supported by commercial RTI interoperation software, MAK-RTI and our own interoperation software, Fed Extender, which is discussed in the later section. Viewed in terms of HLA/RTI, the Exercise Controller is the only time regulating federate in the system. Its most important role is to pace the execution with wallclock time. The Exercise Controller advances logical time in synchrony with wallclock time and other federates synchronize their federate time with the Exercise Controller federate.

Each user can recognize the battlefield within its seek scope as we do in a real battlefield. By recognizing battlefield situation, it can decide its course of action itself or execute a mission of commander direction. Since all users can execute such actions, tactical training comes true in result. All exercise situations from start to stop are comprehensively displayed on 3D synthetic environment which all moving entities, artificial and natural objects are overlapped in visual. Tactical actions such as fire, move, seek etc and damage effect of engagement such as kill and destruction are represented on the synthetic environment. In final, analyzing logged exercise data, AAR can be performed and produce an exercise result in 2D and 3D data format, by individual and whole, by diverse format.

Fig. 8 shows that battlefield is shared with all participants and displayed. The battlefield sharing makes users have common view about battlefield even though their own capability. That makes tactical training possible. It overcomes the limit of individual exercise which only focuses on driving or fire technique exercise. Synthetic battlefield environment of Fig.6 comprehensively represents a situation that a helicopter attacks a tank and a bridge has been destroyed, etc.

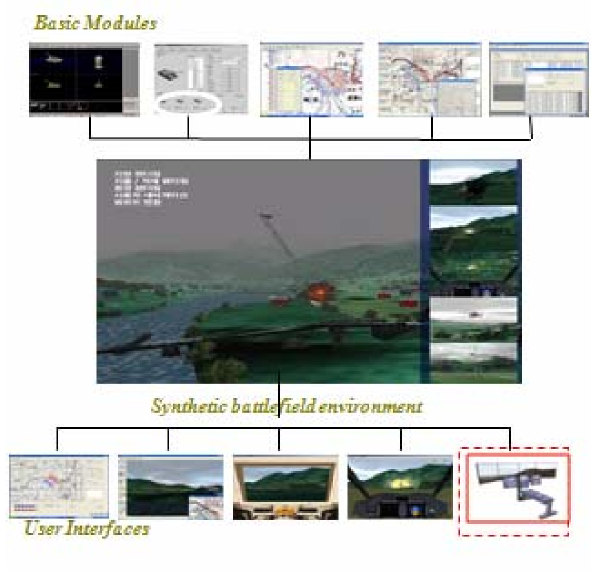


Fig. 8 Battlefield Share and Display

In order to construct environment such as Fig.8, all modules necessary to simulation must be supported. This paper implements the modules of Fig.6 and integrates them into one simulation system, TTSS. TTSS has tested using military exercise scenario by military users and accepted as company-level virtual tactical training simulation system which infantry, weapon (helicopter, tank) and commander can train diverse tactics by interaction with each other on common battlefield.

5 Technical Issues

This section discusses some technical issues which have been identified during TTSS development.

5.1 Performance Estimation for Expansion

Although TTSS is tested at company level tactical training, it is necessary to review whether it may be used for battalion level training since most modules for tactical simulation have been already prepared and may be reused for more large scale exercise. The point is not technique but data. In general, as scale increases, transmission data increases in exponential because entities increases and battlefield expands. This paper estimates the transmission load of battalion level exercise based on the company level transmission rate defined in test. The estimation results are as follows. In case of company-level tactical exercise among infantry, helicopter and tank, the transmission load is about 571.2 Kbps. Expanding to the case of battalion level, it is estimated that transmission load is about 2.23 Mbps, 4 times of that of company-level. This load rate can be adapted enough on local area network. Shortly, in view of number of entities and transmission load, TTSS can afford to use for battalion level exercise as well as company level exercise. Here doesn't discuss this in detail. The detail evaluation and estimation will be described at other papers.

5.2 HLA/RTI based Interoperation

HLA/RTI is a standard for interoperation among heterogeneous defense simulation systems and also IEEE 1516 standard. Many systems have applied to interoperate diverse systems. Many RTI software products have been supplied by COTS, for example, MakRTI, pRTI etc. Although COTS usage increases, the cost for one server with eight clients is about twenty thousands dollars. If users increases, much cost is required for only interoperation. So we introduced the concept of the 'light' RTI version shown at Fig.5, and it has proven to be one of the solutions to reduce the RTI based expansion cost. Federate Extender is simplified RTI software which includes subset of full RTI functions. It doesn't greatly influence to performance and supports expansion. Fig.5 help expand to large scale exercise. In order to enhance to HLA/RTI based interoperation, researches on interoperation among heterogeneous RTIs, real time of RTI, light RTI etc are needed. In a nutshell we can configure our federation in three difference ways: RTI only mode, Federate Extender only mode, and Hybrid mode. The RTI only mode is the genetic method to construct federations. To cover the problems that the RTI only mode has we designed and implemented the two federation modes.

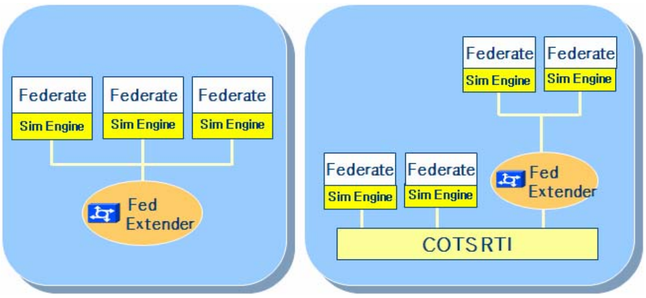


Fig. 9 Federate Extender-only mode and Hybrid mode

As shown in Fig. 9 the Fed-Extender only mode operates federations without Commercial RTI software. By doing so we could both escape from a license problem of commercial software and keep the system performance acceptable. Fig. 10 shows how the Federate Extender mode works. The Federate Extender mode provides executable Fed Extender Server software and the Federate Manager Client (FMC) library for federation developers. With this mode

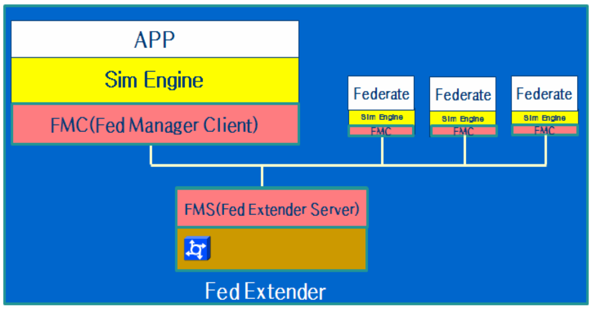


Fig. 10 Fed Extender only mode

The Hybrid mode makes our system peculiar in an HLA community. In the hybrid mode the Federate Extender works as both a federation server and a coordinator of RTI and federates. Fig. 11 depicts the mechanism of the hybrid mode.

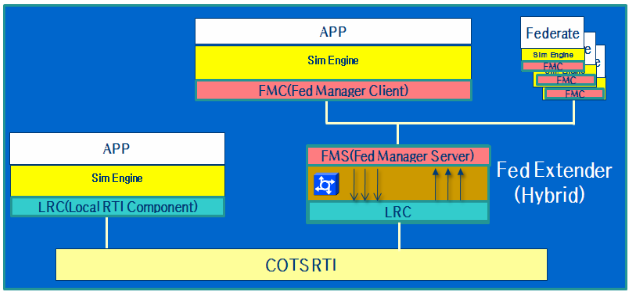
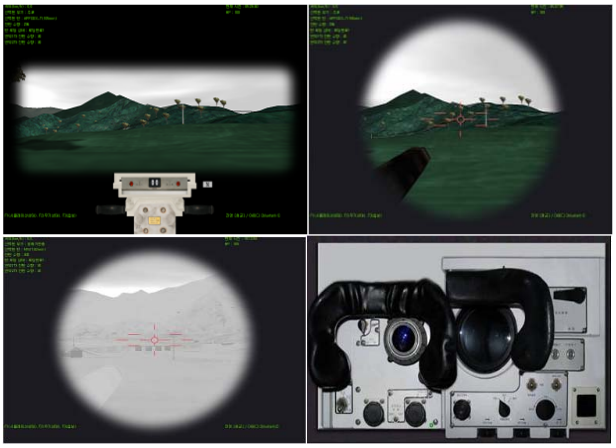


Fig. 11 Hybrid mode

5.3 Reconfigurable Simulation

Differing from existing full simulation, tactical weapon simulator has selectively chosen functions so that it can execute team tactical exercise rather than individual exercise. This paper has designed software based reconfigurable tactical weapon simulators. The display for pilot is changed by software according to weapon model without hardware platform change, for example, from Cobra to Apache helicopter. Fig. 12 shows the display example implemented in this paper. Although reconfigurable tactical simulator doesn't similar to real platform, it is possible to exercise tactics with other opponents and friendly forces on one synthetic battlefield as shown in dot part of Fig 8. Since this simulator type can be constructed by low cost, expansion cost is low and easy. Also since models can be transferred by selecting model display, diverse weapon models can be simulated easily. If diverse models future are models and included in battlefield, any war game may be possible. Software based tactical simulation will be an essential technique for weapon tactical simulation.



a) Tactical tank simulator cockpit

b) Tactical helicopter simulator cockpit

Fig. 12 Software based Reconfigurable Weapon (Helicopter) Tactical Simulator

5.4 CGF: Unmanned Simulation

The technique providing diverse battle situations with friendly or opponent force must be included in tactical simulation. This can be implemented by unmanned autonomous force. Different from manned simulation technique, this technique is a simulation type that a computer itself can recognize battle situation and interact with other entities using own tactical knowledge. It is called Computer Generated Force, CGF. Since CGFs can be run by a computer instead of real users, it is possible to expand exercise scale without real users' participation and even computer system. This is effective in the point of economy and time. As shown in CGF interface of Fig.8, TTSS simulates basic semi-autonomous CGF for a tank and a helicopter that can move along predefined positions and fire with their capability on synthetic battlefield as described in Fig 13. To control and set the rule for our CGF we decided to apply "Lua" script language. Lua is considered as a powerful, fast, light-weight, and embeddable scripting language in an industry area especially in a game development. At the same time Lua is free software, distributed under a very liberal license. It is allowed to use for any purpose, including commercial purposes, at absolutely no cost.

Currently we are planning to enhance autonomy and entities of CGF near future. Certainly, unmanned simulation technique will be an essential technique of dynamic tactic exercise.

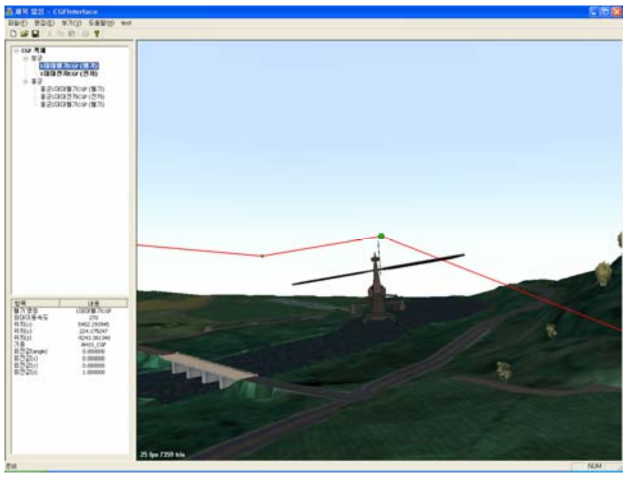


Fig. 13 Helicopter CGF interface (snapshot)

These technical issues identified during TTSS implementation must be more researched in deep and included as essential techniques in VMSF of Fig.3 as follows: Performance issue for expansion and HLA/RTI interoperation enhancement in simulation services layer; Reconfigurable simulation issue and unmanned simulation in simulation models layer.

6 Conclusion

According to level, type, purpose, scale and force, many simulation systems exists. The framework approach is effective method to build up many simulation systems. The framework must have features such as distribution, real time, composibility, interoperability, scalability etc. In particular, for virtual simulation systems, the networked virtual environment technique can be used as a framework.

This paper shows how to construct a tactical training simulation system (TTSS) using virtual modeling & simulation framework (VM&SF) based on networked virtual environment technique. Many other virtual simulation systems in military field can be designed and implemented on VM&SF.

From implementation experience, this paper identifies some technical issues such as performance estimation, HLA/RTI Interoperation, reconfigurable simulation, unmanned simulation etc. These issues must be more enhanced or researched in future. The framework approach will be useful way to engineer many simulation systems in other fields as well as military field.

References

[1] F.Kuhl, R.Wetherly, J.Dahmann, *Creating computer simulation systems-an introduction to the high level architecture*, Prentice Hall PTR, 1999.

[2] M.Lejter, T.Dean, A framework for the development of multiagent architectures, *IEEE Expert*, Dec. 1996, pp. 47-59.

[3] H.S.Sarjoughian, B.P.Zeigler, S.B.Hall, A layered modeling and simulation architecture for agent-based system development, *Proceedings of IEEE*, Vol. 89, No. 2, Feb. 2001, pp. 201-213.

[4] L.F.Wilson, D.J.Burroughs, A.Kumar, J. Sucharitaves, A Framework for linking distributed simulations using software agents, *Proceedings of the IEEE*, Vol. 89, No. 2, Feb. 2001, pp.186-200.

[5] A.I. Conception and B.P Zeigler, "DEVS formalism: a framework for hierarchical model development, IEEE Tran. On Software Eng. Vol 14, No. 2, pp.228-241, 1988

[6] EEE1516.2-2000, IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA)-Object Model Template (OMT) Specification, 2000

[7] Rita Simons, John W. Schaefer, Jim Melzer, *Visualization Solutions for AVCATT-A Reconfigurable Cockpits*, I/ITSEC, 2002

[8] William R. Johnson, Thomas W. Mastaglio, Paul D. Peterson, *The Close Combat Tactical Trainer Program*, Proceedings of the Winter Simulation Conference, 1993