Proceedings of the 1994 Winter Simulation Conference
ed. J. D. Tew, S. Manivannan, D. A. Sadowski, and A. F. Seila

SYNCHRONIZING SIMULATIONS IN DISTRIBUTED INTERACTIVE SIMULATION

Sandra Cheung
Margaret Loper

Institute for Simulation and Training
3280 Progress Drive
Orlando, Florida 32826

ABSTRACT

Distributed Interactive Simulation (DIS) provides the infrastructure to build large-scale simulations by interconnecting independent simulators via a network. These simulators execute primarily in real time because of the human-in-the-loop requirements. The future of DIS in the general area of distributed simulation will include non real time simulations, such as constructive wargames. Active research in the area of Parallel Discrete Event Simulation (PDES) has devised means by which event driven simulations can be parallelized, by utilizing state of the art parallel architectures and scheduling events which can be executed concurrently and in a conflict-free fashion. Simulation has experienced growth in both these communities. Each community has specific needs and requirements which have to be met. This paper examines the areas of DIS in which PDES techniques may be employed to achieve more parallelism. In particular, this paper addresses the need for, and implementation of synchronization in current DIS networks. The interaction between constructive and other categories of simulation will require synchronized DIS applications, and employ mechanisms utilized in PDES.

1 INTRODUCTION

1.1 Distributed Interactive Simulation

Distributed Interactive Simulation (DIS) is an infrastructure that enables heterogeneous simulators to interoperate in a time and space coherent environment. In DIS, the virtual world is modeled as a set of entities that interact with each other by means of events that they cause. Simulator nodes independently simulate the activities of one or more entities in the simulation and report their attributes and actions of interest to other simulator nodes[1]. The simulator nodes are linked by a communication network and communicate entity information using a set of common network protocols.

An essential part of the DIS infrastructure is a series of standards in the areas of interface definition, communication, management, security, field instrumentation, and performance measurement (DIS Steering Committee 1994). To date, most of the work has been in the definition of information (protocol data units or PDUs) communicated between simulations to make them interoperable. These PDUs have been assembled into a standard which was approved by IEEE in March 1993 (IEEE 1278 1993). Subsequent versions of the protocol standard are in development (Institute for Simulation and Training 1994b).

Another DIS standard, communication architecture (Institute for Simulation and Training 1994a), specifies the requirements for the underlying network (see Figure 1) in a DIS. The most notable requirement is real-time delivery (100 to 300 milliseconds) of the protocol messages to the simulation nodes on the network.

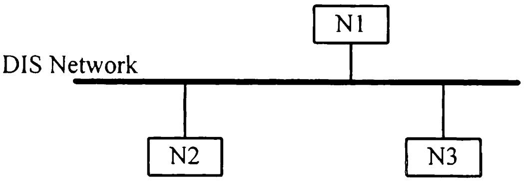

Figure 1. DIS Network

In DIS, there is no central computer. Instead, a number of computers are interconnected via a network (such as the one shown in Figure 1). DIS is

---

[1] As a point of reference, DIS is often used to simulate virtual battlefields, where the simulated entities are tanks, infantry fighting vehicles, combat aircraft, and infantry.

based on a number of design principles (DIS Steering Committee 1994).

- **Autonomy of Simulation Nodes.** Each node is only responsible for the entity or entities it is simulating, and does not have to calculate what other nodes are interested in. Receiving simulations are responsible for determining the effects of an event on the entities it is simulating. The autonomy principle enables nodes to join or leave an exercise in progress without disrupting the simulation.
- **Transmission of "Ground Truth" Information.** Each node transmits the absolute truth about the state of the entity/entities it simulates. The receiving nodes are solely responsible for determining whether their objects can perceive an event and whether they are affected by it. Degradation of information (essential for realistic portrayal of system behavior) is performed by the receiving nodes.
- **Transmission of State Change Information Only.** Simulations will only transmit changes in the behavior of the entities they represent, in order to reduce unnecessary information exchange.
- **Dead-Reckoning Mechanisms.** The objective of dead-reckoning (a term borrowed from navigation) is to determine new states based on previous ones, i.e. by extrapolation. Only when the ground truth data differs enough from the extrapolated data (by a predetermined threshold) is a new state issued.
- **Simulation Time Constraints.** Current DIS standards primarily support human-in-the-loop simulations. The simulation time constraints (100 - 300 milliseconds) were obtained based on human factors. Other types of simulations (such as wargames) operate faster or slower than real time. In order for these types of simulations to interact with real time simulations, interfaces to the constructive simulation need to be capable of issuing data at real time rates.

Categories of Simulations

DIS is intended to support a mixture of simulation types: virtual, live, and constructive. The historical core of DIS has been continuous, real-time simulations, which have been designated as "virtual." These simulations include human-in-the-loop or crewed simulators and computer generated forces (CGFs). CGFs can populate the environment with a large number of entities under loose supervisory control of human operators. Virtual simulations are characterized by their requirement for real-time delivery, which is on the order of 100 milliseconds (Institute for Simulation and Training 1994a).

DIS is also intended to interface with "live" simulations which include operational platforms and test & evaluation systems. Interactions between real weapon systems, sensors, and tactical communication links occur at much faster rates than virtual simulations, often less than one millisecond.

The last type of simulation is event driven wargames, called "constructive" simulations. Constructive simulations differ from the other two in that the level of simulation is at a higher level than that of a single entity. These simulations often move faster or slower than real-time. The intervals at which the states of all the participants are updated may be irregular and minutes may elapse between them.

1.2 Parallel Discrete Event Simulation

PDES is another form of distributed simulation, and refers to the execution of a single discrete event simulation program on a parallel machine. PDES research has really taken off since the early 80s, driven by the computational demands of large discrete-event simulations and the availability of high-performance computers. PDES programs benefit very little from other parallel techniques such as vectorization on supercomputers.

In a discrete event simulation, the system being simulated changes state only at discrete points in time, upon the occurrence of an event. PDES schedules events which can be executed in a concurrent fashion. These events occur at different simulated times, but introduce synchronization problems, known as causality errors (Fujimoto 1990).

PDES mechanisms fall into two broad categories: conservative and optimistic. Conservative methods avoid situations in which causality errors can arise. Optimistic methods use a detection and recovery mechanisms. Upon detection of a causality error, a rollback mechanism is employed to recover.

Despite the methods employed by PDES to speed up the execution of discrete event simulations, the impact on the general simulation community has not been significant (Fujimoto 1993).

1.3 Fundamental Differences in DIS and PDES

In general, DIS and PDES are based on different premises, namely, distributed versus shared memory and networked environment versus single host. These differences are depicted in Figure 2.

PDES is based on a shared memory paradigm. One computer simulates the entire system. While events might be distributed across multiple processors, there is no autonomy -- decisions regarding the simulation state are made in a central location. Also, PDES is a single host environment in that it is not geographically distributed over a network. While PDES does not suffer from communication latency, it can suffer from synchronization problems.

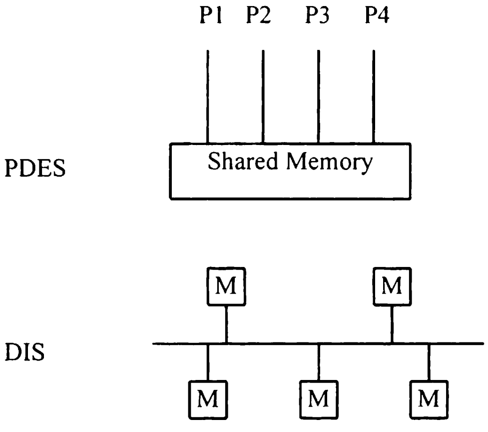

Figure 2. PDES and DIS Environments

With the advent of DIS, simulation has moved to a distributed environment, both in memory and geographical location. This new technology brings with it new problems to solve. Instead of a simulation residing on one computer, the simulation resides on multiple computers, each responding to events in the virtual world. This environment is heterogeneous and can be effected by issues such as communication latency and time synchronization.

For example, the DIS Protocol Standard (Institute for Simulation and Training 1994b) specifies the circumstances under which PDUs are transmitted and received. The simulation applications, responsible for issuing PDUs for the entities they are simulating, should never issue PDUs in advance of the occurrence of the event or state (no post-dating). However, due to communication latency, the PDUs received from external simulated entities may arrive out of order, and the simulation application is responsible for accommodating for this. In addition, the receiving simulations must be able to determine whether the incoming PDUs contain more current information than that currently being used to model the transmitting entity. If so, the application must update its model using the information contained in the PDU, otherwise the PDU is to be discarded.

1.4 Paper Organization

The remainder of this paper is organized as follows. In Section 2, the need and motivation for having synchronization in DIS are described. Section 3 describes DIS for non real time applications. Section 4 addresses the implementation and complexity issues of synchronizing DIS networks. Section 5 describes aggregate level simulations, and in particular two case studies are provided (ALSP and Eagle/BDS-D). Finally, Section 6 presents some concluding remarks.

2 MOTIVATION

The motivation for synchronization in DIS comes from the way in which the technology will be used. The initial focus of DIS-based applications has been on training, especially the training of large, joint, or combined forces. Closely associated with training is mission rehearsal, in which essential coordination procedures are worked out and practiced. DIS is also envisioned to support Test & Evaluation (T&E), where one can do initial prototype evaluations and rehearse field tests to save valuable test range time. Each of these uses will require some level of synchronization to ensure that hardware clock differences, communication delays, and host computer loads do not skew or corrupt critical exercise data. This is especially true in large-scale simulations where hundreds of dissimilar simulation nodes, with multiple software units, will be used to create the virtual battlefield.

In the sections that follow, several examples are described which identify the need for synchronization. It is important to note that the examples vary from operational (planning an exercise) to analytical. For each use, a different degree of synchronization will be required. This multi-level synchronization is indicative of the multiple uses of DIS.

2.1 Latency

As stated earlier, (Institute for Simulation and Training 1994a) specifies a latency requirement of between 100-300 milliseconds. This latency relates to the issue of how tightly a simulated entity is coupled to the entity to which it is reacting. To illustrate, consider interfacing virtual and live simulations to evaluate a new weapons system. Linking a virtual missile with a live mid-course guidance system can add confidence to test results and verify system integration prior to the first live firing. For this type of tightly coupled interaction, where latency must be less than a millisecond, synchronization of the simulations is critical, especially since test results will be the basis for further design criteria as well as operational procedures for the fielded system.

2.2 PDU Ordering

In a DIS exercise, simulation nodes may be co-located via Local Area Networks (LANs) or distributed over large geographical distances, and connected via Wide Area Networks (WANs). Latency incurred in LANs is negligible; however, WANs can induce large latencies in the transmission of data. If this happens, PDUs may arrive at destination nodes out of order or not at all. For general entity information, this is not a problem as one of the key design principles dictates periodic state updates. However, when critical events such as weapons fire occur out of order or are missed, it can cause problems with the simulation. This is especially true if the simulation node missing the PDU models the entity under fire. There are several solutions to this problem (reliable transfer, simulation back-up) and an accurate time reference will be critical part of the answer.

2.3 Simulation Management

From an operational viewpoint, DIS will be used for joint service training. In order to accomplish this, an accurate common time reference will be required (Figart 1991). Each service will preplan their exercise with time controlled events being triggered by a common scenario script. This type of simulation management is essential in order to provide the synchronization demanded to conduct a realistic exercise using the real-time Command, Control and Communications (C3) systems available to the services today.

2.4 Simulation of Weapons Systems

From a training perspective, simulations of time sensitive actions such as electronic emissions and missile launch will require an accurate common time reference (Figart 1991). Electronic Warfare (EW) simulations require strict coordination based on radar scan rates, antenna heights, as well as power output and emitter characteristics to ensure that an emission is received when the radar antenna is directed toward the receiving unit. Realistic simulation of these types of EW exercises require the finest coordination between the units involved tending to indicate a need for an accurate time source. Missile launch exercises also require close coordination between the units involved. Target detection times, lock-on times, launch and kill times must be recorded with an accuracy within milliseconds to permit proper evaluation of the exercise.

2.5 After Action Review

From an analytical perspective, time synchronization is critical to post experiment analysis. In order to reconstruct a scenario or exercise event for analysis, it will often require using data from more than one simulation node. Reconciling the differences between different system clocks can be a problem. To accomplish this, an accurate time stamp on each protocol data unit will be necessary in order to extrapolate data to its proper time reference. For example, analyzing the exact time and distance of line-of-sight, weapons fire, and detonation in a scenario is important for applications such as mission rehearsal and virtual prototyping.

From a communications perspective, time synchronization is required in order to correlate simulation events with bandwidth utilization. For example, correlating entity collisions or EW interactions with peak bandwidth rates will help communication designers size networks for DIS exercises.

3 NON REAL TIME DIS

The main purpose of DIS is to provide heterogeneous, geographically dispersed simulators interaction in the same synthetic environment. The training community has benefitted tremendously from this technology which reduces the cost of planning, rehearsing and training large scale forces. The simulators which are traditionally part of these exercises are considered real time because of the requirements for man-in-the-loop training.

With the advent of a wider variety of distributed applications, DIS is no longer confined to have a real time "look and feel", but will incorporate simulations which execute faster or slower than real time. This is an objective that was described in Section 1.1.

Non real time DIS applications include scenario review and debugging, concept review, and detailed analysis (Miller 1994). Training in a non real time environment is the target of the Above Real Time Training (ARTT) program (Guckenberger 1994). Other applications of non real time DIS is the use of logistics. This will play an important role in analysis and interpretation purposes (Douglass 1994). Current DIS exercises simulate time periods which range from minutes to a few hours. In these cases, logistics will hardly have an impact as it affects simulations with time periods of weeks, months or even years.

4 IMPLEMENTATION ISSUES

The set of PDUs specified by DIS and described in (Institute for Simulation and Training 1994b) all consist of a global PDU header, which among other pieces of information contains a field reserved for the timestamp. According to (Institute for Simulation and Training 1994b), the timestamp should indicate the time at which the data in the PDU is valid. Simulation applications must therefore insert a valid timestamp (obtained from some time source) prior to transmitting the PDUs. These PDUs also incur a latency due to the network, which may be negligible on a LAN, but may non-trivial on a WAN. This delay can be determined by the difference between the originating and arrival timestamps, assuming all sites on the network were synchronized.

This section describes the methods by which systems on an internet can synchronize within a reasonable accuracy. For DIS environments, the minimum accuracy has not yet been established, and is thus left as a parameter of the specific application or as an open problem.

4.1 General Approaches

There are several ways in which time servers can synchronize their clocks to the Coordinated Universal Time (UTC). There are radio services which broadcast the UTC timecode modulation which can be decoded by suitable receivers. These timecode receivers can be quite costly, and thus providing each site with such a device would prove impractical.

The U.S. National Institute of Standards and Technology (NIST) provides a dial-up time service, which transmits the time code via modems. This service is good for infrequent synchronization of individual systems.

Reliable clock synchronization can also be achieved using agreement algorithms (Lamport 1985) but these algorithms do not necessarily produce the most accurate and precise time, and may produce unacceptable network overheads in a large network (Mills 1990).

4.2 Global Positioning System (GPS).

The NAVSTAR GPS is developed and deployed by the U.S. Department of Defense, and to date the most accurate radio navigation system. One of its uses is for the distribution of precise time and frequency. GPS consists of a 24 satellite constellation, each of which carries a set of atomic clocks which are tracked and maintain UTC to better than 100 nanoseconds. The complete system contains 6 orbits of four satellites, each at a height of 10,900 miles, thus ensuring that any spot on the earth will have at least 4 visible satellites at any moment.

A GPS receiver tracks the spread spectrum code of 4 satellites simultaneously. By measuring the time of arrival of the signals, the time of year, and position is obtained. Once the position of the receiving antenna is known, only one satellite is required to determine and maintain timing accuracies.

Examples of some large scale distributed simulations which use GPS receivers are the Real Time Clock System (RTCS) developed by the U.S. Army Space and Strategic Defense Command's Advanced Research Center (Kress 1994) and Battle Force Tactical Training (BFTT) developed by the Naval Surface Warfare Center (Forbes 1994).

4.3 Network Time Protocol (NTP).

NTP is a protocol which is designed to distribute standard time using the Internet. To synchronize a distributed system using NTP, a set of primary reference sources (synchronized directly by external reference sources such as timecode receivers) are set up to be the time servers for other stations on the network (which act as their clients). In order to provide time reliably, the network can be configured in such a way that when primary time servers become unavailable, secondary servers can provide the time (Mills 1990).

The accuracies achievable by NTP depend on the precision of the local-clock hardware and stringent control of device and process latencies. The advantages of NTP is that the protocol can be easily implemented, has been ported to multiple Unix platforms, and causes relatively low network load (and hence is not a burden to normal network operations). However, clock synchronization may take long periods of time and multiple comparisons are needed in order to maintain accurate timekeeping (Mills 1992). The accuracy is directly dependent on the time taken to achieve it.

4.4 Synchronizing DIS Systems over a Network

In (Institute for Simulation and Training 1994a), the Communications Architecture working group of the DIS community has ensured that the underlying media, types of services, and protocols are common and meet DIS performance requirements. (Institute for Simulation and Training 1994a) also specifies that DIS systems are to be synchronized by means of NTP.

There are several topologies which have been proposed to achieve this objective (Katz 1994). The distinction between a LAN and a WAN become essential for the reasons stated in Section 2.2.

- Equip each DIS node on the network with a reliable time server. This method will achieve the maximum accuracy, and synchronization will not cause additional network traffic, but is impractical due to the cost.
- Equip each LAN with a time server. The signal can be propagated to each node on the network by a dedicated network (and thereby not causing interference with DIS traffic) or share the network over which other traffic, including DIS, is transmitted. The latter configuration would subject the timing packets to additional delays incurred by sharing the medium with other network traffic, but can be easily implemented using NTP. The placement of the time server on the LAN in either case is independent of the DIS nodes.
- Place the time server at the gateway between the LAN and the WAN. The gateway is then responsible for synchronizing with the remote nodes, by placing the timestamps on outgoing PDUs. For incoming PDUs, the gateway uses the timestamp to extrapolate the state information to the current time, and then transmits these modified PDUs over the LAN. The assumption made here is that delays over a LAN are negligible.

5 AGGREGATE LEVEL SIMULATIONS

As was described the Section 1, simulations can be divided into three categories: live, virtual, and constructive. The primary difference between the latter and the former two types is that in constructive simulation, the granularity of entities being simulated is much higher. The level of simulation used both in live and virtual simulation is at the level of individual entities which are operating in real time. In contrast, a constructive simulation set of entities is grouped into an "aggregated unit", which is then simulated as a single object. Constructive simulations typically operate in non-real time, i.e. faster or slower than real time.

The DIS environment has traditionally consisted of entity-level simulations. These types of simulations have provided a capable means for training soldiers, and for evaluating new weapons and vehicle systems. But in order to scale to 100,000s entities, other solutions must be evaluated. One such solution is to use computer generated forces, that can populate the environment with minor human control. Another method is to use aggregate level simulations (DIS Steering Committee 1994). Aggregating entities has the advantage over CGF entities in that the network bandwidth requirement is less (PDUs describe entire units rather than the individual components).

Currently, there are several projects researching the interconnection of constructive and virtual simulations. The Aggregate Level Simulation Protocol is investigating the interconnection of constructive simulations, and Eagle/BDS-D is looking at interfacing constructive and virtual simulations. Each is described below.

5.1 Aggregate Level Simulation Protocol (ALSP)

ALSP simulations are exercises involving a number of simulations called actors. These actors can interact by sending messages describing the state of the entities which they control. An actor can thus be considered an aggregate unit. The ALSP architecture can be seen in Figure 3. The ALSP Common Module (ACM) is an element which must be attached to each actor. A grouping of actors and ACMs in the same exercise is also known as a confederation. Messages destined to the confederation are first transmitted to the ACM, and similarly messages from the confederation are first received by the ACM. Communication between an actor and its ACM is bi-directional and point-to-point. ACM to ACM communication uses a broadcast/multicast mechanism.

Actors coordinate the advancement of simulation time with the confederation through the ACM. Each actor has its own simulation time (which is the value it has last requested from the ACM). The confederation time is a set of simulation times of the actors represented in the confederation. In order for an actor not to receive messages from the past, the ACM queues all messages in timestamp order. When an actor requests an advancement in time, the ACM first sends it all the messages which have a timestamp less than or equal to the requested time, before it grants the actor the advancement (Mitre Informal Report 1992).

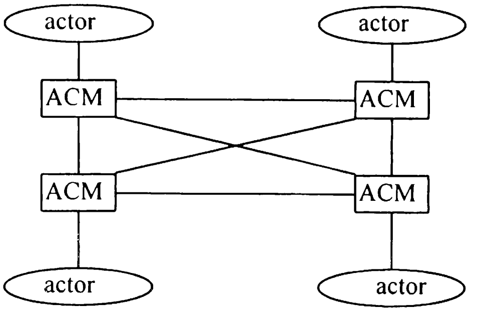

Figure 3. ALSP Architecture

When new actors join the confederation, other actors either catch up to its simulation time (if it is ahead) or are forced to wait until the new actor catches up (if its simulation time is "in the past") (Mitre Informal Report 1992).

5.2 Eagle/BDS-D

The Eagle/BDS-D project was designed to prove that an aggregate simulation (the Eagle combat model) was capable of interoperating with an entity level simulation, such as a CGF. Eagle is a corps/division level combat model that simulates ground combat at the company and battalion level, and runs faster than real time (Karr 1994).

The movement, position, and status of aggregated units are maintained for the unit as a whole, and are the result of statistical analysis of the unit's actions rather than the result of those of the individual entities (Karr 1994).

Aggregated Eagle units can disaggregate into individual components, and the execution is shifted to real time. In the disaggregated form, interaction between other entities in the environment, such as those from a CGF, can occur. Aggregated units may also interact with entity level simulations. A disaggregated Eagle unit can request to be reaggregated upon finishing interaction. A Simulation Integration Unit (SIU) is responsible for the disaggregation and aggregation of units, and maintains the composition of the disaggregated units (Karr 1994). The difficulties between the interaction of aggregated units and entities arise because of the entity granularity and the time scales in which these two types of simulations execute. The SIU of the Eagle model has similar functions as the ACM in ALSP.

6 CONCLUSION

This paper presented an overview of the design principles of DIS. In particular the need for synchronization is addressed from several perspectives. DIS deploys primarily real time, entity level simulations which interact with one another by exchanging standard messages (PDUs). The agreement made among the various simulations is that no PDU will be postdated, and that the simulations should be synchronized according to a global clock (time server). A timestamp field is provided in each PDU, though the correct use of it is currently not implemented. There are several methods by which distributed systems can synchronize their local clocks within a certain range. However, the accuracy to which they are to be synchronized has not been determined by the DIS standard.

Finally, the paper discusses two case studies of aggregate level simulations and how they deal with synchronization and interaction with real time simulations (as found in DIS). These methods fall into the conservative PDES approaches. Since DIS attempts to allow a seamless integration of the three categories of simulation, the interaction and interoperability between virtual and constructive simulations is still an active area of investigation, and perhaps other PDES mechanisms should be employed to solve synchronization problems, and to further enhance parallelism.

REFERENCES

DIS Steering Committee. 1994. *"The DIS Vision, A Map to the Future of Distributed Simulation (Version 1),"* IST-SP-94-01.

Douglass, W. M. 1994. *"The Impact of Time on Simulation,"* Proceedings of the 10th DIS Workshop on Standards for the Interoperability of Defense Simulations, Volume II, pp. 381-396.

Faye, J. P. and D. Taylor. 1994. *"Variable Scaling Time Implementations in DIS,"* Proceedings of the 10th DIS Workshop on Standards for the Interoperability of Defense Simulations, Volume II, pp. 397-404.

Figart, G. 1991. *"Time Synchronization,"* Proceedings of the 4th DIS Workshop on Standards for the Interoperability of Defense Simulations.

Forbes, J. 1994. *"Synchronization and Absolute Time Stamping in the DIS Environment,"* Proceedings of the 10th DIS Workshop on Standards for the Interoperability of Defense Simulations, Volume II, pp. 625-628.

Fujimoto, R. M. 1990. *"Parallel Discrete Event Simulation,"* Communications of the ACM, Volume 33, No. 10, pp. 31-53.

Fujimoto, R. M. 1993. *"Parallel Discrete Event Simulation: Will the Field Survive?,"* ORSA Journal on Computing, Volume 5, No. 3, pp. 213- 230.

Guckenberger, D., E. Guckenberger, and D. Mapes. 1994. *"Virtual Time: Reducing Stress and Temporal Workload,"* Proceedings of the 10th DIS Workshop on Standards for the Interoperability of Defense Simulations, Volume II, pp. 417-426.

IEEE 1278. 1993. Standard for Information Technology -- Protocols for Distributed Interactive Simulation Applications.

Institute for Simulation and Training. 1994. Standard for Distributed Interactive Simulation -- Communication Architecture Requirements, IST-CR-94-14.

Institute for Simulation and Training. 1994. Standard for Distributed Interactive Simulation -- Application Protocols, Version 2 Fourth Draft, IST-CR-94-50.

Karr, C. R. and E. Root, 1994. *"Integrating Aggregate and Vehicle Level Simulations,"* Proceedings of the Fourth Conference on Computer Generated Forces and Behavioral Representation, pp. 425-435.

Katz, A. 1994. *"The Absolute Clock in the DIS Scheme,"* Proceedings of the 10th DIS Workshop on Standards for the Interoperability of Defense Simulations, Volume II, pp. 1-4.

Kress, J., J. R. Phipps and D. Carver, Jr. 1994. *"Synchronization of Large Scale Distributed Simulations and Programs,"* Proceedings of the 10th DIS Workshop on Standards for the Interoperability of Defense Simulations, Volume II, pp. 611-624.

Miller, J. 1994. *"Time Scaling of Simulations in a DIS Environment,"* Proceedings of the 10th DIS Workshop on Standards for the Interoperability of Defense Simulations, Volume II, pp. 223-232.

Mills, D. L. 1990. *"On the Accuracy and Stability of Clocks Synchronized by the Network Time Protocol in the Internet System,"* ACM Computer Communication Review Volume 20, Number 1, pp. 65-75.

Mills, D. L. 1992. *"Network Time Protocol (Version 3) Specification, Implementation, and Analysis,"* RFC-1305.

MITRE Informal Report. 1992. Aggregate Level Simulation Protocol -- Technical Specification.

AUTHOR BIOGRAPHIES

SANDRA E. CHEUNG obtained her B.S. from Florida Institute for Technology, and Ph.D. from the University of Florida in 1988 and 1993 respectively. Her research interests include performance evaluation, traffic characterization, high speed networks, interconnection networks, and parallel architectures. Since joining IST in 1993, she has been involved in DIS compliance testing, analysis of DIS traffic patterns, and performance evaluation of DIS architectures.

MARGARET L. LOPER is Program Manager for the DIS Standards project at the Institute for Simulation and Training. As chair of the DIS standards committee, she is responsible for the development of the DIS standards and their progression through national and international standards organizations. Ms. Loper earned a M.S. in Computer Engineering from the University of Central Florida and a B.S. in Electrical Engineering from Clemson University. Her research interests are in simulation and communications.