Proceedings of the 1998 Winter Simulation Conference
D.J. Medeiros, E.F. Watson, J.S. Carson and M.S. Manivannan, eds.

BOUNDING CPU UTILIZATION AS A PART OF THE MODEL DESIGN AND THE SCENARIO DESIGN OF A LARGE-SCALE MILITARY TRAINING SIMULATION

William R. Merritt

Lockheed Martin
12306 Lake Underhill Rd MP 855,
Orlando, FL 32825-5002, U.S.A.

ABSTRACT

Goals of large-scale training simulations include scalability, compose-ability, and extensibility in affordable open systems architectures. System designs that employ, Optimistic Time Warp, predictive contracts, interest expressions, and distributed systems methods offer great potential to accomplish this goal. However, the combination of these concepts produces an extremely complex and dynamic solution space. This paper demonstrates a method that can aid in both the design decisions for the level of detail of models, and the composition of scenario models. A basic simulation model that approximates one node of a highly interactive distributed military training simulation with platform level entities is developed. The entities will be effected by events that represent fire, sensing, and command organization. The simulation outputs will then be used to develop regression equations. The regression equation predicts performance of scenarios within the experimental region.

1 INTRODUCTION

Future large-scale military training simulations need to support Unified Commands, Joint Task Force, Training Effectiveness, Readiness, Planning, Interoperability, and Exercises (Worley 1996) for organization and planning in response to a rapidly emerging crisis. A system design capable of supporting train on that level requires a highly complex set of interactions, but neither at one instance in time, nor for every scenario. The heterogeneity of the interactions spread over time can be exploited in a discrete event simulation to better utilize the CPU resources and network bandwidth that must be employed to support large-scale military training simulation. Time Warp (TW) (Fujimoto 1990), Breathing Time Warp (Steinman93), and High Level Architecture (DMSO), Data Distribution Manager (DDM) (Cohen 1998) and (Van Hook 1998) filtering (Bassiouni 1997), offers a tremendous opportunity to direct the computational power of a Wide Area Network (WAN) towards the solution that is needed for the variations in the specific levels of model detail that is needed. But nothing comes for free. The prevalent method of testing a simulation is to run it and see if it works. For a specific scenario, the entity models can or can not be distributed to maintain real-time response to an operator for a given set of hardware. Scenario composition decisions must be made. The simulation capabilities must be sufficiently flexible to support composition decisions.

Because of the cost and size of a training exercise for brigade and above commanders, better methods of evaluating the dynamic complexity in a training scenario need to be employed. The methods exist. We only need to apply the methods to model design, and scenario generation. Then answers for questions such as "How will a scenario analysts know that the simulation will execute in real-time?" can be quantified and answered to some confidence level.

In Optimist Time Warp simulation virtual time can advance as quickly as events are resolved by object methods. When real-time operator interaction is an essential part of the simulation, the Time Warp method provides the opportunity for simulation to complete before the synchronization with the real-time operator. Therefore the computational requirements of an operator command can be separated in virtual time from the bulk of computation. For the virtual time the command is entered, only those events must be satisfied to maintain a real-time view of events to the training audience. Characterization of event effects in the simulation with regression equations gives a quick way to rate the impact of operator decisions.

2 THE POWER OF PREDICTION

Movement and sensing have been the largest consumers of CPU and network resources. A predictive contract (Mellon 1995) expresses the physical property of an entity's behavior that may be represented by a function for a specific period of simulation time. The predictive contract's function provides an interacting model with data at a specified time to determine the property of interest and compute the interaction. DIS uses predictive contracts, such as Dead Reckoning, to reduce network traffic. Extending the concept of a predictive contract can further reduce network traffic. The model in this paper uses linear B-splines (Bartels 1987) with 4 segments as the route model for movement. Whereas DIS protocol defaulted to posting one heading and an another either every 5 seconds or if direction varied by more than 10 degrees. Interest Expressions (Powell 1996) complement predictive contracts. Whereas predictive contracts state a function about the entity's behavior, interest expressions define the attributes of interest to the entity about other simulated entities, such as location or frequency. An Interest Manager (Mellon 1995) (HLA) in the infrastructure can then minimize computation between entities as well as the network load by restricting unneeded invocation of entities. Multi-cast (Powell 1996) is an identification of the specific sets of nodes that need to receive an entity's update based on interest expressions.

3 ORIGINS OF CONCEPTS

The High Level Architecture (HLA) addresses many of the lessons learned about scalability. Predictive Contracts (Mellon 1995), Interest Expressions (Powell 1996), and multi-cast groups (Mellon 1996) are some of the most intensely powerful concepts that will play a significant role in the success of 100,000+ entity simulations. These methods will form the basis of scalability in open system architectures. The DIS Dead Reckoning constraints were simple and allowed for easy addition to a simulation of another simulator. However, as the number of players increased so did the network utilization until either the network was saturated or the CPUs of the players were saturated because of filtering unneeded entity state updates. CPU availability to perform entity modeling reduced inversely to the time spent receiving a PDU from the network and deciding if the PDU was needed on that node. Although advances in network hardware have increased the network bandwidth such that the network constraints have relaxed, the increase in CPU utilization that is needed to filter data and control data in a distributed simulation is a central issue of the HLA Data Distribution Manager (DDM). Key techniques of filtering, predictive contracts, interest expressions, and multi-cast groups, are combined with a Time Warp Mechanism (Fujimoto 1990) such as Breathing Time Warp for the Joint Simulation System (JSIMS) project. This combination of methods offers a great potential for the scalability in open systems architectures of large-scale military training simulation. Bringing these methods together offers a tremendous opportunity to leverage training goals with computational and network constraints. Optimistic Time Warp with lazy cancellation offers the advantage of advancing simulation virtual time (VT) ahead of the real-time clock and only needing to compute the changes brought about by operator interaction.

3.1 Time Warp DDES and Model Detail

Optimistic Time Warp with lazy cancellation (Fujimoto 1990) assumes that the logical processes (LP), the part of a simulation that produces useful work, can independently execute to completion or to some VT limit. Causality errors are the result of LP interacting at different virtual times. Rollbacks are the mechanism that is used to return the interacting LP with the greatest time to the LP with the lowest time. The underlaying assumption is that interactions in the simulation do not form a single sequence of events that depend on the outputs of the previous event. A logical process both receives and produces event messages. The objective is to use the inherent autonomy of LPs and gain computational speed from parallelism. In the case of large-scale military training simulations the parallelism comes from distribution of the simulation over Local Area Networks (LAN), and Wide Area Networks (WAN). The logical processes of military simulations correspond to either single entities or groups of entities. To receive the benefit of parallelism from DDES time warp, the distribution of entity models must be accomplished such that highly interactive entities are allocated to one node of the network. Allocation to one node reduces the number of interactions that suffer delay from network latency. Extensive resolution of interactions over the network can reduce the advance of useful work to a conservative time management and completely negate the desired benefit of distribution. The entire simulation could simply advance from one event to the next without any parallelism achieved. Load balancing becomes a function of organizing the simulated entities in a convenient grouping on network nodes to minimize the number of rollbacks introduced by the latency in the network.

Another way of expressing this for an entity in a constructive simulation is "What is a sufficient level of model detail to achieve the training effect and maintain the advance of virtual simulation time to the training audience?" This view of design criteria shifts the focus from a single level of detail about an entity model to "What is the rate of virtual time advance that can be achieved by this mix of models as a function of computation and frequency of interaction?" An example, supply convoys do not need the same level of detail in a behavioral model as a "keyhole killer". A keyhole killer is a scout with an excellent point of view to direct precise artillery fire onto targets. This can significantly alter force ratios The supply convoy model can make a predictive contract over a greater VT period than a scout vehicle finding a location of partial concealment.

3.2 Filtering

Work has been done on demonstrating that linear computational complexity, O(N), of 2-D dynamic sectoring (Harless 1995) and 3-D dynamic sectoring (Paulo 1997) can be approached. Sectoring and hierarchical filtering methods of proximity detection (Steinman 1995) require dynamic adaptation of the sector size. Predictive contracts (PC) have already been shown to reduce network updates (Mellon 1995). Filter spaces (Mellon 1996) formed by interest expressions (Powell 1996) coupled with predictive contracts and replicated objects (Agrawal 1992) will contribute to successful large-scale simulations. However, the complexity introduced by these concepts must be understood to establish effective control over the inherent interactions of a 100,000+ entity military simulation. No simple method exists to analyze the complexities. The following model will demonstrate the analysis techniques to investigate the bounds of a feasible solution space for a large-scale military simulation.

3.3 Useful Work in DDES

Useful work (Palaniswamy 1994) defines constraints that can be adapted for the purpose of evaluating models under Time Warp. The essential concept of useful work that will be used here is the computation, or event resolution, that becomes permanent with the advance of global virtual time (GVT) (Fujjimoto 1990). Useful work is a function of the ratio of virtual time advance to computational time. A ratio of 1/1 defines a real-time (RT) environment. Time Warp allows for specific models to have a ratio of less than one and still maintain an effective rate of 1:1 as long as the event is not continuous and the frequency allows complete resolution prior to GVT. With the constraint of useful work in mind, the question: how do you determine the useful work measure during design and during scenario composition? Useful work, initially, defined a measure for the scheduling order of logical processes to computational resources. This paper presents a combination of methods to improve our insight into a simulation scenario's scale and composition such that a real-time rate of GVT advance can be maintained under external, operator initiated interaction.

4 THE EXPERIMENT

For a large-scale military simulation and the purposes of this paper assume that an entity has one predictive contract and at least one interest expression. A predictive contract defines the behavior of an entity over a period of virtual time. In the DIS community, Dead Reckoning effectively reduced network updates. Predictive contracts that hold true for a longer virtual time further reduce the frequency of network updates. In a Time Warp DDES time is necessary attribute. At an expiration time, a new virtual time relevant predictive contract must be made and distributed to the network. The interest expression provides the data to a filter mechanism in the network infrastructure (Mellon 1995) to determine where both passive updates and event messages must be distributed. Filtering methods include hierarchical filtering, and proximity detection, and sectoring. Hierarchical filtering (Mellon 1996) as used in the Joint Precision Strike Demonstration (Powell 1996) was deemed successfully and identified that the size of the segment was most important to the success. Proximity detection (Steinman 1994) indicated that the size of the grids was most important. Sectoring (Harless 1995; Paulo 1998) studies indicate that with dynamic sector adjustment of the size of the sectors **O**(N) complexity can be obtained. Because of the previous work on filtering method, the following simulation model assumes that near linear computational complexity will be used in large-scale military simulation and will not be addressed in this paper.

4.1 Scope of Model

The most controllable and observable portion of a large-scale simulation distribution is a single network node and the node's response to simulation events from the network. This study produced a simulation model that characterizes interactions between entities on one node and the arrival of events from other sources in the distributed simulation. The interactions have two responses; a new predictive contract is computed, and the update of the entity is transmitted to the rest of the simulation. Bounds are selected for a full factorial experimental design (Cavitt 1996). The simulation results are used to determine a regression equation for CPU utilization. The regression equation functions to approximate interaction effects between level of detail in design and the composition of a scenario. The objective is to establish a method that aids in design decisions for software and hardware choices. The combined analysis techniques of simulation and design of experiments provide a powerful tool (Law and Kelton 90; Hood 1993; Donohue 1994; Sanchez 1994) for approximation. This collection of methods can build confidence that a large-scale training simulation composition of models and hardware will be successful. In this context, success is defined as maintaining an overall rate of advance of simulation time as the constraint GVT/RT >= 1. The constraint GVT/RT>=1 was used to determine the upper bounds on the model.

4.2 The Simulation Model

The model is designed to maintain near real-time operator interaction. The simulation consists of one node on a network. The node assumes a Optimistic Time Warp operating system with lazy cancellation and replicated objects. The basic model is accomplished by having each entity post a predictive contract that expires at some virtual time. When expiration occurs, a new predictive contract must be computed and the new entity description is made available to the distributed system. Event messages by unique categories arrive at different rates. The effect of the event is that some of the entities are selected and caused to compute a new predictive contract. The CPU time to compute a new contract is collected. The number of updates to the network is collected. This is sufficient to demonstrate the complexity of the factor interactions that exist in such a system.

4.3 Factors Defined

A simplification of a military simulation involves number of entities, movement, firing, seeing and being seen, as well as responding to commands. Entities will represent the number of significant objects that assigned to the processing node. Immediate replacement of entities is assumed for this simulation. Movement will be represented by the CPU time required to calculate a new predictive contract. Firing, sensing, and commands come in event messages. Event messages define an interaction event. Interaction event types have a frequency and an extent. Frequency is measured by how often an event can occur. Extent is the probability that an entity is affected by the event. It is significant to note that for later models each one of these factors can be defined to greater levels of detail and specialization, but that assessment is relegated to future work. The model will establish high and low values for each of these 5 factors. The terms low and high are used to identify the order of the numerical value of the factor in the experiment not the effect. In terms of frequency a low frequency means less frequent than a high frequency, but the low factor is simply the lower of 2 values. For simplicity the 5 factors will be called entities, cpu cost, fire, sensing, and organization, respectively. The simulation model will be run for 3 replicates for both an outer and inner full factorial design with center points using the values for the factors in tables 1, 2, and 3.

Table 1: Outer Factorial

Factor	**Low**	**High**	**Extent**
Entities	1000	4000	none
Cpu Cost (seconds)	.001	.006	none
Fire (seconds)	5.001	25	.01
Sense (seconds)	5.001	30	.025
Organization (seconds)	15	60	.06

In addition to the boundary simulation run an additional set of simulations was run for an inner factorial experiment with center points to provide additional data for second order analysis. Other designs such as a central composite design could be used for second order analysis. The significant issue in deciding the design approach is cost. In this specific circumstance selecting a full factorial design within the region of the first factors ranges was not prohibitive in either time or resources. The low and high values for the inner factorial are contained in Table 2.

Table 2: Inner Factorial

Factor	Low	High	Extent
Entities	2000	3000	none
Cpu Cost (seconds)	.00225	.00375	none
Fire (seconds)	10	20	.01
Sense (seconds)	11.25	18.75	.025
Organization (seconds)	26.25	33.75	.06

For the center points, the factor settings are illustrated in Table 3.

Table 3 – Center points

Factor	**Center**	**Extent**
Entities	2500	none
Cpu Cost (seconds)	.0035	none
Fire (seconds)	15	.01
Sense (seconds)	17.5	.025
Organization (seconds)	37.5	.06

The low and high factors fire, sense, and organization are used as the means for uniform distributions that varies by -5.0 and + 5.0 seconds. The uniform distribution is used for 2 reasons. The first reason is that an event can occur anywhere within that region. The second reason is that the range provides variability about effects that may result as an operator's command or response to the simulation. The range around the factor also eliminates any bias that can be introduced into the simulation model by having the events arrive at perfectly synchronous rates (Law and Kelton 1990). The most frequent that an event can arrive is limited by the network latency. The latency reflected in this model is 0.001 seconds which corresponds to the max rate of PDUs that were observed in the 50,000 ModSAF project (Brunett 1998). The minimum CPU cost was also selected as 0.001 to reflect that the lowest CPU cost was no better than network latency. The high CPU cost matches the cost of a position update from the Close Combat Tactical Trainer (CCTT).

4.3.1 Fire

The category of fire includes close combat, direct fire, and indirect fire. Since an event is carried out at the entity level, the distinction is not significant for this demonstration. For the scope of the current analysis, no entities are declared inactive. The model assumes replacements are immediately available. All entities compute a new predictive contract. The predictive contract is the movement model as described in the section of this paper called "Movement Model".

4.3.2 Sensing

Sensing in this model combines response, or electronic counter measures, to being sensed as well as sensing the immediate surroundings as a result of an entity state change. Since the passive updates of all entities are distributed to the network. We assume that the large area sensors are on a separate node and can complete their assessment in a completely autonomous manner.

4.3.3 Organization

The organization accounts for command and control orders that require immediate reaction of the entities. Again the movement model accounts for the CPU utilization of the command and control orders.

4.4 Movement Model

Each entity is assigned a route prior to the beginning of the simulation. The CPU utilization for a predictive contract computation and traversal is drawn by selecting N points in x, y pairs from a uniform distribution over the interval of 1 to 1000. The predictive contract represents a route that has N –1 segments. If N is greater than 2 then the route is a linear B-spline (Bartlels 1987). Also an entity speed between the ranges of 0.5 and 8.5 m/s is selected and applied to the distance of the route to calculate the next predictive contract. The distance/speed is the time that the predictive contract expires. The CPU time utilization for that entity is the (distance in kilometers) * (the cpu cost) for a the respective experiment (see table 2). The randomness adds stochastic variation to the CPU time utilization of the entity as well as dispersing the expiration of the predictive contracts. All entities are active at some frequency to compute their next predictive contract. It is important to note that since each entity is unique, as more specialization information to the design detail of the models is available more differentiation between entities is possible. The stochastic variation should be maintained, unless, of course, the design forces synchronization on the entities. Synchronized entities would be the case where more than one entity is modeled under one logical process (LP). In a scenario in which no interaction occurs on an entity, a new predictive contract for the entity only needs to be computed when the existing one expires.

4.5 Event Effects

An event effect is a function of the frequency and the extent of the event. The events arrive according to the frequency of the respective factor**.** Extent provides limit on the engagement size. The model selects a number of entities, K, to be affected by the event. K is drawn from a uniform distribution with a range of 0 to number of entities on the processor. The total number of entities that will be affected by that event will be K * extent. Extent is a probability value. The settings of fire, sensing and organization are 0.01, 0.025, 0.06, respectively. The numbers that were chosen reflect a limited engagement scenario. The number of entities affected by fire and sensing event ranges from a company to companies but only battalion sized groups were affected by organization events at the high factor level for entities.

4.6 Simulation & Design Of Experiments

The simulation could be run for any combination of the defined inputs to determine the effect. However, this would only provide information for that one set of initialization values. Design of Experiments (DOE) (Law and Kelton 1990; Hood 1993; Cavitt 1996) provides a systematic way to examine a feasible region of the solution space. Systematic analysis of the factors at selected levels can reveal the significant interactions in the system. This simulation used only 5 factors. So a full factorial design that requires 32 separate simulation runs with specific settings of the factors was reasonable to use. Since the simulation approximated one hour of real time in about 5 minutes, the number of experiments to run for an inner and outer full factorial with center points and 3 replications was possible. So it was done. As the number of factors of the simulation increase, fractional factorials can be used for initial screening of significant factors, and factor interaction. Regression analysis of the response of the simulation runs will identify the significant terms of a regression equation. The regression equation could then be used in place of the simulation model (Law and Kelton 1990) as long as the values of the factors in the regression equation stay within the ranges of the experiment. With Response Surface Methodology (Myers 1995), techniques such as variance reduction, and gradient estimation can then be employed to better understand the effects in the response brought about by small changes in the input factors. But these methods will not be addressed in this paper.

4.6.1 Factorial Design

The design of experiments (Myers 1995) is as follows where a "–" indicates a low level, a "+" indicates a high level, and a 0 indicates a center level of the factor. For each run 3 replicates were run and random number sequence was not restarted. The first run with the outer factorial did not include the center points. For the second run the random number stream was advanced past the end of its last seed from the first run.

4.6.2 Execution

The length of run of the model was selected to reflect one hour to the training audience. The criteria of GVT/RT >=1 was used to judge that the simulation results are in the feasible solution space. One hour was used as the length of time at which the test GVT/RT >= 1 was made. Sensitivity analysis was performed on the number of segments in a route. The highest value for the number of segments that met the requirement GVT/RT criteria was 5. All routes were computed to contain 4 segments. At this point each experiment was ran for 3 replicates at each setting of the factors for both the outer, and inner full factorial DOE. Also 3 replicates were ran at the center point of the experimental region. The total was 195 simulation runs. Care was taken to continue the random seed (Law and Kelton 1990). This provided ample coverage of the experimental region for a second order regression equation.

4.6.3 Discussion of Simulation Results.

The CPU time utilization ran from the low value of 0.79% for experiment run 7, up to the high value of 69.57% for run 25. The center runs and the interior full factorial all lay within these extremes. Regression analysis was conducted. Regression analysis consists of iterative regression of the sample data and removing the terms with the highest p-value until the lowest Mean Square Error (MSE) is found. The lowest MSE for CPU Utilization is 0.000842172. The terms that remained significant at a p-value of 0.05 contribute to the response. The statistic R^2 and adjusted R^2 are 96% and 95% respectively. The R^2 statistic indicates the predictive capability of the regression model. The reduced quadratic model adequately represents the CPU utilization for the region of the experiment. The regression analysis results that were obtained are illustrated in Table 4.

5 SUMMARY

This paper demonstrates the feasibility of approximating the resource utilization of a large-scale simulation with the combination of a simulation model, Design Of Experiments and regression analysis. The only requirement put on the design of the target large-scale simulation models is the identification of CPU utilization weight for the object models of the entities in the military simulation models. The complete set of event interactions are known as a function of software design. The interaction between objects is discretely designed as part of the system. Once a simulation exercise is planned the level of model detail is constrained to the behavior, or resource consumption, of the objects modeled in that scenario. Likewise, the set of object interactions is completely known. The interactions can be segregated into two categories; the preplanned interactions as a result of mission operations, and the training audience operator initiated interaction during the exercise. Unlike time-step that attempts to maintain only the present time, and optimistic Time Warp based methodology can maintain the resolution of preplanned simulation interaction in advance of the real-time perspective of a training audience operator. So the question of "How much CPU resource is needed?" is a function of the object interaction and the CPU utilization of those object methods that resolve the event. Therefore with simple instrumentation of methods during development, or deployment of a simulation object model, the CPU utilization weight of methods can be known. Further division of the simulation exercise into epochs, either fixed or variable duration time intervals of study, can isolate the set of events that must be resolved during that epoch in the planned mission group. This defines a base level of CPU utilization for that epoch on that processor. This paper has demonstrated how to construct such predictive equations. If we extend this method then regression equations that estimate the CPU resource that is need as a consequence of training audience commands is also feasible to develop.

Table 4: Regression Analysis Results

```
      The regression equation is
      CpuAMean = - 0.0501 +0.000053 Entities - 0.0122 Sensing + 0.00128 Organization
           + 37.7 CpuCost -0.000002 Entities*Sensing  -0.00000049 Entities*Organization
           + 0.0180 Entities*CpuCost - 0.177 Fire*CpuCost
           - 1.42 Sensing*CpuCost  - 0.350 Organization*CpuCost +0.000501 Sensing*Sensing

              Coef       Stdev    t-ratio        p  Predictor
          -0.05009     0.01906      -2.63    0.009 Constant
        0.00005252  0.00000537       9.78    0.000 Entities
         -0.012224    0.001350      -9.05    0.000 Sensing
         0.0012759   0.0003048       4.19    0.000 Organization
            37.654       3.472      10.85    0.000 Cpu Cost
       -0.00000202  0.00000016     -12.96    0.000 Entities*Sensing
       -0.00000049  0.00000009      -5.62    0.000 Entities*Organization
         0.0180399   0.0007791      23.16    0.000 Entities*CpuCost
          -0.17657     0.06308      -2.80    0.006 Fire*CpuCost
          -1.41753     0.09206     -15.40    0.000 Sensing*CpuCost
          -0.35046     0.05110      -6.86    0.000 Organization*CpuCost
        0.00050102  0.00003529      14.20    0.000 Sensing*Sensing

      s = 0.02902     R-sq = 96.1%     R-sq(adj) = 95.9%

      Analysis of Variance

      SOURCE       DF          SS          MS         F        p
      Regression   11     3.79354     0.34487    409.50    0.000
      Error       183     0.15412     0.00084
Total       194     3.94765
```

With these condition recognized thresholds can be determined that can be employed to signal to either simulation technical support of automatic systems that actions need to be taken to maintain the real-time advance of the simulation exercise. Further investigation into this concept will be the subject of future work.

ACKNOWLEDGEMENTS

Thanks to Carla Powe, Program Manager, and Robert Knox, System Architect, of the WARSIM 2000 program for the opportunity to investigate performance requirements for large-scale military simulations.

REFERENCES

Agrawal, D., Agre, J.R., 1992. Replicated Objects in the Time Warp Simulations. In Proceeding of 1992 Winter Simulation Conference. 657-664.

Bartels, R.H, Beatty, J.C. and Barsky, B.A. 1987. Introduction to Splines for use in Comptuer Graphics & Geometric Modeling. Morgan Kaufmann, Los Altos, CA.

Bassiouni, M.A., Chiu, M., Loper, M., Garnsey, M. and Williams, J. 1997. “Performance and Reliability Analysis of Relevance Filtering for Scalable Distributed Interactive Simulation.” In ACM Transactions on Modeling and Computer Simulation,Vol.7, No. 3, July 1997, 293-331.

Brunett, S. and Gottschalk, T., 1998. “Scalable ModSAF Simulations With More Than 50,000 Vehicles Using Multiple Scalable Parallel Processors.” In 1998 Spring Simulation Interoperability Workshop. 98S-SIW-181.

Cavitt, D.B., 1996. “A Performance Analysis Model for Distributed Simulations.” In Proceeding of the 1996 Winter Simulation Conference. 629-636

Cohen, D., and Kemkes, A. 1998. DDM Scenarios. In 1997 Fall Simulation Interoperabiltiy Workshop. 97F-SIW-057

DMSO. Defense Modeling and Simulation Office, “DoD Modeling and Simulation Master Plan” http://www.dmso.mil/docslib/mspolicy/msmp/1095msmp/Chap2.html Oct95 DoD 5000.59-P

Donohue, J.M. 1994 “Experimental Designs for Simulation.” In Proceeding of the 1996 Winter Simulation Conference. 200-205.

Fujimoto, R.M. 1990. “Parallel Discrete Event Simulation.” Communications of the ACM, Vol 33 (70):30-53

Fujimoto, R.M., 1993. “Parallel and Distributed Discrete Event Simulation: Algorithms and Applications.” In Prodeeding of the 1993 Winter Simulatin Conference. 106-114

Harless,G. and Rogers, R.V., 1995. “Achieving O(N) in Simulating the Billiards Problem in Discrete-Event

Simulation." In Proceedings of the 1995 Winter Simulation Conference. 751-756.

HLA. see DMSO

Hood, S.J. and Welch. P.D. 1993. "Response Surface Methodology and its Application in Simulation." In Proceedings of the 1993 Winter Simulation Conference 115-121.

Law, A.M. and Kelton W.D., 1990, Simulation Modeling and Analysis (Second Edition). McGraw-Hill, New York, NY

Mellon, L. and West. D., 1995. "Architectural optimizations to Advanced Distributed Simulation. "In Proceedings of the 1995 Winter Simulation conference. 634-641.

Mellon, L. Spring 1996. Hierarchical "Filtering in the STOW System. " 14th Workshop on the Interoperability of Distributed Interactive Simulation. <u>http://www.sisostds.org/dis/dis-library/index.cfm</u> [96-14-087]

Mellon, L.F., 1996. "Intelligent Addressing and Routing of Data Via the RTI Filter Constructs." 15th Workshop on the Interoperability of Distributed Interactive Simulation. 96-15-122

Myers, R.H. and Montgomery, D.C. 1995. Response Surface Methodology: Process and Product Optimization Using Designed Experiments. John Wiley & Sons, Inc. , New York, NY

Palaniswamy, A.C. and Wilsey, P.A. 1994. "Scheduling Time Warp Processes Using adaptive Control Techniques." In Proceedings of the 1994 Winter Simualtion Conference . 731-738

Paulo, E.P. and Malone, L.C., 1997. "Methodology for the Increase Computational Efficiency of Discrete-Event Simulation in 3 Dimensional Space." In Proceedings of the 1997 Winter Simulations Conference. 525-531.

Powell,E.T., 1996. "The use of Multicast and Interest Management in the DIS and HLA Applications." In ." 15th Workshop on the Interoperability of Distributed Interactive Simulation. 96-15-37 257-266.

Sanchez, S. M., 1994. "A Robust Design Tutorial." In Proceedings of the 1994 Winter Simulation conference. 106-113

Stalcup, B. 1998. "High Fidelity Distributed Simulation Analysis." In 1998 Spring Simulation Interoperability Workshop. 98S-SIW-105

Steinman, J.S. 1993. "Breathing Time Warp" In Proceedings of the 1993 workshop on Parallel and distributed simulation. 109-118.

Steinman, J.S., 1994. "Parallel Proximity Detection and the distribution List Algorithm", In Proceedings of the 1994 workshop on Parallel and distributed simulation. 3-11

Van Hook, D.J. and Calvin, J.O. 1998 "Data Distribution Management in RTI 1.3" In 1998 Spring Interoperablility Workshop. 98S-SIW-206.

AUTHOR BIOGRAPHY

WILLIAM R. MERRITT is a systems engineer for Lockheed Martin Information Systems in Orlando FL. He received a B.S. degree in computer science from North Carolina State University and is currently engaged in obtaining a M.S. from the industrial engineering department from the University of Central Florida. PTL.