# Modeling And Simulation Environment Server For Distributed/embedded Environment Representation: Final Report

1-1-1999

Glenn A. Martin

Guy Schiavone

INSTITUTE FOR SIMULATION & TRAINING

Author: Martin, Glenn; Schiavone, Guy
Title:
Subj: Report
Subj; simulation
Subj; dynamic environment
Subj; : simulation graphics
Contract # W99007
IST # ISTCR-99-05

**CONTRACT W99007**

**December 29, 1999**

# FINAL REPORT

Date Dec 29, 1999

University of Central Florida

IST-CR-99-05

# Modeling and Simulation Environment Server for Distributed/Embedded Environment Representation

*Final Report*

**Contract W99007**

December 29, 1999

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

IST-CR-99-05

INSTITUTE for SIMULATION & TRAINING

SUBMITTED TO:

CONCEPTUAL SYSTEMS & SOFTWARE
1174 HIGHWAY 80
CHINO VALLEY, AZ 86323

MODELING AND SIMULATION ENVIRONMENT SERVER FOR DISTRIBUTED/EMBEDDED ENVIRONMENT REPRESENTATION

CONTRACT W99007

SUBMITTED BY:

INSTITUTE FOR SIMULATION AND TRAINING
UNIVERSITY OF CENTRAL FLORIDA
3280 PROGRESS DRIVE
ORLANDO, FL 32826-0544

GLENN A. MARTIN
GUY SCHIAVONE
CO-PRINCIPAL INVESTIGATORS
407-658-5000
{martin, guy}@ist.ucf.edu

# Modeling and Simulation Environment Server for Distributed/Embedded Environment Representation

## Final Report

**Glenn A. Martin and Guy Schiavone**
**Institute for Simulation and Training**

**December 29, 1999**

## Abstract

In the past, simulators have used static environments. Only in recent years has a more dynamic environment been of interest to simulation builders. This report documents an effort by Conceptual Systems & Software, Inc. and the University of Central Florida's Institute for Simulation & Training investigating adding a dynamic environment server to the Battle Force Tactical Training program. Specifically, focus was given to two components of this program: JointSAF and the Generic Navy Stimulator/Simulator. In addition, the Tactical Environmental Data System and the Navy Integrated Tactical Environmental Subsystem were also studied due to the desire to connect with these systems and access their data.

Building a dynamic environment server for the Battle Force Tactical Training program requires investigation of three components: defining an architecture, defining the environment services, and defining the data structures. Actually, each of these components builds on top of the latter ones. Therefore, the greatest power resides in the data structures themselves that are used to represent the enumerated services that in turn affect the design of the architecture. The most powerful data representation to date has been splines. However, while they have had numerous advantages, a major disadvantage has been inaccurate interpolation of data points. The recent building of non-oscillatory splines has solved that problem. In addition, the recent advances in wavelet theory have provided an additional means of data compression to reduce bandwidth and memory use. These technologies together can be used to build a very powerful, robust and flexible distributed database that can be used in a bottleneck-free distributed system.

# 1.0 Introduction

For many years, simulator systems focused on the vehicles in use and neglected the environment. They used simple static environments that could not affect the vehicles actually being used in the simulation. However, in recent years there has been a slow increase of interest in adding a more dynamic environment to simulations. To date this has largely involved changes to more visible environment data such as terrain elevation (exceptions include physical models from the Institute for Simulation & Training and the U.S. Army Waterways Experiment Station). However, there are "non-visible" data that also represent the environment and can affect entities (such as soil cohesion or air moisture).

## *1.1 Purpose*

Under this project, the Institute for Simulation & Training (IST) at the University of Central Florida assisted Conceptual Systems & Software (CSS) in research towards the development of a dynamic environment server for distributed and embedded systems. Specifically, this research was performed within the context of the U.S. Navy's Battle Force Tactical Training (BFTT) program. The project had the following overall tasks:

- Evaluate and Define Applicable Simulation Model
- Explore Alternative Concepts for E-Server Processing
- Define the Distributed E-Server Architecture
- Give a Proof-of-Concept Prototype Demonstration
- Deliver the Phase I Final Report

On July 19, 1999, a meeting was held with Dick Esslinger, Doug Clark and John Stortz of the BFTT office at the Naval Sea Systems Command in Arlington, VA. At this meeting, the goals of the project were re-defined and it was agreed that CSS and IST would focus on just a pair of the trainers in use within BFTT and how the two trainers could be modified to use an environment server. In particular, we agreed to investigate a design for the environment server integration for JointSAF (JSAF) and the Generic Navy Stimulator/Simulator (GNSS). In addition, it was also agreed that any prototype work should be performed in Phase II. The new goals for the project then became:

- Define an E-Server Architecture
- Define Environmental Services
- Define Required Data Structures

This report discusses each of the tasks above in the context of the three new goals. The report itself fulfills the last task.

## *1.2 Dynamic Environment Experience at IST*

IST has participated in a number of projects related to dynamic environments. They have included terrain, fluid and building dynamics as well as synthetic environment representation and correlation. Specifically, these projects include:

- *Dynamic Terrain* – In this three-year project, innovative new software architectures were developed to support dynamic terrain and fluids. To support a more extensible terrain representation that can provide a variety of interactions, alternatives to the traditional polygonal representation were explored. Models for vehicle tracks, craters, mines, bulldozers, and a mine-clearing breacher were completed. Completed software is available to DoD contractors. The work was funded by U.S. Army STRICOM and the Defense Modeling and Simulation Office (DMSO) in collaboration with USATEC, USAWES, AES and NRL.
- *Dynamic Buildings* – As part of the Marine Corps' Team Tactical Engagement Simulator (TTES), IST developed data structures and algorithms for real-time modifications of building walls based on simulated breaching charges. The polygonal representation of a building was dynamically re-worked in order to represent the resulting hole. This project was sponsored by NAWC-TSD.
- *Weapons Effects for DIS Applications* – As part of an SBIR with Visidyne, Inc., IST developed algorithms to dynamically alter a general-purpose terrain to place craters. The important effort of this project was the application of the cratering algorithm to any database, not just one formatted in a particular fashion. This project was sponsored by the Defense Special Weapons Agency.
- *SIF Enhancements and API Development* – The goal and result of this projects was to enhance the Standard Simulator Data Base Interchange Format (SIF) and develop a robust SIF database interchange mechanism to facilitate cost-effective, efficient archiving and re-use of USAF (and other) simulator terrain database assets. The interchange mechanism involves the development of a comprehensive, non-proprietary software access interface to SIF. The project was sponsored by the Air Force.
- *ZCAP Software Development* – ZCAP (Z-Correlation Analysis Program) is a suite of software tools that addresses terrain database interoperability issues in networked interactive simulation. Currently, it provides capabilities for terrain correlation testing, culture correlation testing, line of sight (LOS) correlation testing, coordinate conversions, shift detection, terrain database sampling, terrain remeditation, and terrain database visualization. It allows for import from several data formats including SIF, OpenFlight, S1000 and CTDB. The project was funded by U.S. Army STRICOM.
- *Benchmarks for Low-Cost Visualization Platforms* – IST is investigating uses for emerging PC graphics products in simulators. IST recently signed key agreements with Microsoft and the OpenGL Performance Committee, and has several arrangements pending with graphics suppliers. The agreements provide ways to insert simulation graphics requirements into commercial products and standards very early in the development cycle. PC products generally provide lower cost solutions than dedicated graphics products. In its laboratories, IST can evaluate strategies to

insert simulation requirements into products by working with suppliers and procurement organizations. IST believes these relationships will form a new paradigm for quickly developing new products and standards, and provide for further enhancements in the performance of PC graphic systems. This project was funded by U.S. Army STRICOM.

- *Dynamic Terrain Synthetic Environment Server* – IST assisted Conceptual Systems & Software in developing a dynamic terrain server for simulations hosted on a low-cost PC platform. A prototype breacher simulation was developed on a PC with positive results. This effort was sponsored by U.S. Army STRICOM.

## 2.0 Define an E-Server Architecture

Due to the U.S. Department of Defense requirement for all systems to be compliant with the High-Level Architecture (HLA), the BFTT program also wishes to move towards HLA. Any environment server built for BFTT must operate within HLA. In order to be HLA compliant, there are a number of rules that must be followed, but of major importance are the development of a Federation Object Model (FOM) for the federation and a Simulation Object Model (SOM) for each federate.

A FOM for BFTT will define the objects, attributes and interactions both to be allowed and to be shared within the federation. In order to complete this, a survey of the environment services and data structures needed is required. This survey can be found in the following sections of this report.

The definition of a SOM for each BFTT is a complex task but is used to define the role and capabilities of each component. The BFTT program contains a large collection of legacy simulators and stimulators that have been more or less pieced together to work as the BFTT system. Furthermore, each component has been built separately and ownership resides with each individual vendor. Some of the components are hardware-based systems while others are software-based systems. Moving each component to HLA using a FOM developed for BFTT will be difficult, but not impossible.

Once the FOM and SOMs have been designed and a system running on top of the Run-Time Infrastructure (RTI) is built, there is still the question of architecture. The BFTT program is somewhat unique in that it will require multiple levels of operation: battle force training in port, unit level training in port, and battle force training at sea. In other words, each federation execution may have some subset of federates participating and this subset may not be constant across multiple federation executions. This causes a scenario where the environment server will need to exist in multiple locations depending on the current operation being performed. However, in order to understand the ramifications of this issue, we must examine the environment services required first.

## 3.0 Define Environment Services

As mentioned above, during this Phase I SBIR, focus was given to the JSAF and GNSS components of BFTT. While information on JSAF is difficult to obtain, it is known that it is ModSAF-like. In addition, the BFTT program would find it advantageous to have the ability to interface with the Tactical Environmental Data System (TEDS) and the Navy Integrated Tactical Environmental Subsystem (NITES).

However, even more important than specific data to be represented is what different *classes* of data need to be represented. As with any software system, proper abstraction and encapsulation can provide a very powerful mechanism for implementing various key data representations. This includes specifics such as data structures (which will be discussed in the next section), but also includes more general concepts such as:

- *Pre-distributed, pre-computed data.* This data has long life spans that will not change significantly during a BFTT exercise and can exist as simple table lookups during an exercise.
- *Pre-distributed, run-computed data.* This data exist as models local to each federate that are distributed before the exercise and run during the exercise generating environmental data.
- *Run-distributed, pre-computed data.* This data might be tables that are distributed during the exercise and really is of no use during a simulation (it is better to pre-distribute them as well).
- *Run-distributed, run-computed data.* This data exist as models but only one model exists per each federation (rather than per each federate).

These concepts describe *where* data lives and *how* it gets distributed, but it still does not mention *what* it looks like. During the STRICOM Dynamic Terrain project, IST developed a system called the Shared Environment that represented 2-D data meshes as spline surfaces. This provided a generic representation for 2-D data within the database; however, it did not function as a generic representation for data transmission. Instead, a gridded post scheme was used within the defined network protocol data unit in order to maintain a general scheme useful for all vendors. Similar approaches will be needed for 3-D data.

## 4.0 Define Required Data Structures

In defining data structures, it is important to keep in mind that two types of general data will be represented: 2-D and 3-D. As mentioned, IST built a 2-D system built upon cubic spline surfaces. However, this approach did have a limitation caused by the smoothing of the spline surface over the data points (a spline surface may not always interpolate through actual data points but rather define a close approximation of the data points). This limitation can be improved with the addition of more data points but this will increase the memory requirements necessary to represent the data. Two technologies have been developed in the recent years, however, that solve this problem: non-oscillatory splines, and wavelets.

### 4.1 Non-oscillatory Splines

Non-oscillatory splines refer to a new class of splines developed by Dr. John Lavery at the Army Research Office for use with terrain representation [1, 2]. A group led by Dr. David Gilsinn at the National Institute of Standards and Technology is currently evaluating their specific use on quadtrees and triangulated irregular networks. These splines provide smooth and shape-preserving fitting of data (including abrupt changes) while avoiding the "oscillation" of splines that cause data points to be missed within an interpolation. This makes them an ideal representation for data (whether 2-D or 3-D). In addition, non-oscillatory splines maintain accuracy even across coarse resolution data. They certainly will be of important use to environmental representation in the future.

### 4.2 Wavelets

During the 1990's, the use of wavelets has been thoroughly researched [3, 4]. In particular, they are very well suited for data and terrain compression. The heart of wavelet theory is a process called multiresolution analysis. To understand this process, consider a signal $C^n$ that is simply a column vector of $m$ pieces of data $[c_1,...,c_m]^T$. Now, suppose a lower-resolution version of $C^n$ is required (call it $C^{n-1}$) that only has $m'$ ($m' < m$) data items. To compute this, we simply use a form of filtering and downsampling:

$$C^{n-1} = A^n C^n$$

where $A^n$ is a filter matrix that is multiplied with $C^n$.

Clearly, some information is lost in the downsampling. However, if $A^n$ is chosen in a certain way, we can capture the lost detail as another column vector (call it $D^{n-1}$). Another filter matrix $B^n$ (which is related to $A^n$) with dimensions of $(m-m') \times m$ is used to find the lost detail:

$$D^{n-1} = B^n C^n$$

The filter matrices and together are called analysis filters, and the process of splitting $C^n$ into the two parts $C^{n-1}$ and $D^{n-1}$ is called decomposition.

We can also take this process one step further. If $A^n$ and $B^n$ are chosen in a certain manner, then the original $C^n$ can be recomputed from $C^{n-1}$ and $D^{n-1}$ through another pair of matrices $P^{n-1}$ and $Q^{n-1}$ :

$$C^n = P^n C^{n-1} + Q^n D^{n-1}$$

$P^n$ and $Q^n$ are referred to as synthesis filters and the process of recomputing $C^n$ is known as reconstruction.

The process of decomposition can be applied repetitively to produce a hierarchy of lower-resolution signal and detail coefficients. This process is known as a filter. The signal composed of the elements of $C^0$, $D^0$, $D^1$, ..., $D^{n-1}$ is known as the wavelet transform of the original signal $C^n$.

Describing how these sets of analysis and synthesis filters are developed is beyond the scope of this document, but there is a large supply of choices when deciding on filters to use. Indeed, some are even based on using cubic splines, which make them meld well with the new non-oscillatory spline technique.

## 5.0 Connecting it All Together

The BFTT program requires the representation of 2-D and 3-D environmental data. In the general case, splines are very well suited for such representation. The creation of non-oscillatory splines eliminates the one setback caused from the use of splines. In addition, wavelet transforms can further aid in bandwidth and memory consideration by allowing compression of the data points required to properly define a spline surface representing the environmental data.

The environment required by BFTT (or even in the specific case of GNSS, JSAF and TEDS/NITES) all fall within the classes of 2-D or 3-D data. Thus, each independent environment variable can be represented by a 2-D or 3-D spline system. A dynamic environment server will simply contain all the layers required for environment representation. In the future if an additional variable is required, an additional layer of data can easily be added to the database contained within the environment server.

As mentioned earlier in this document, the environment server containing all the various layers of data can run as a federate in an HLA federation. Depending on the various subsets of BFTT components currently in use (such as in port vs. at sea), the environment server can be executed on an appropriate station. However, a bottleneck at what amounts to a central server could cause unsatisfactory performance. When needed, multiple environment servers may be required (possibly one per battle force) which will necessitate a careful transaction protocol between them. This distributed nature of the environment database creates concurrency issues; however, these problems are solvable whereas the bottleneck at a central server is not.

## 6.0 Conclusions

From studying the BFTT program, it is clear that an environment server is a logical next step for it. Unfortunately, the components of BFTT are largely legacy simulators that have been connected with LAN access units. In addition, many components are total hardware solutions. This makes the integration of an environment server more difficult, but not impossible since the vendors of each component retained ownership.

In order to represent a diverse set of environment data, it is prudent to step back and look at possible underlying representations that can support multiple types of data. In addition, having an abstract concept of the data allows the powerful mechanism of adding layers of data in the future (perhaps as each BFTT component is modified to use the new dynamic environment server). Cubic splines were always a valuable asset for this use. With the addition of non-oscillatory splines, the major issues with splines are avoided which allows them to be the ideal representation for all 2-D and 3-D data. Recent work in wavelet theory has provided significant progress in methods for more compactly representing environmental data.

Adding a dynamic environment server to the BFTT program consists of two major components: data structures and architecture. The power of non-oscillatory splines and wavelet theory provides the data structures. HLA provides a high-level answer to the architecture, but issues with concurrent transaction control will inevitably come up during any Phase II work.

## References

[1] http://www.amso.army.mil/amip00/TERRAIN3.htm

[2] E-mail exchange with Dr. John Lavery.

[3] Martin, Glenn A. Multiresolution Analysis of Curves and Surfaces for Use in Terrain Representation. M.S. Thesis, University of Central Florida, Spring 1995.

[4] Michael Lounsbery and Tony D. DeRose and Joe Warren. Multiresolution Analysis for Surfaces of Arbitrary Topological Type, ACM Transactions on Graphics, 16(1), pp. 34-73 (January 1997).