Towards Cross Domain Terrain Services

Jesse Campos, Steven Borkman, Gregory Peele, Chuck Campbell
Applied Research Associates
Orlando, FL

jcampos@ara.com, sborkman@ara.com, gpeele@ara.com, ccampbell@ara.com

ABSTRACT

The One Tactical Engagement Simulation System (OneTESS) is pushing the bounds of simulation performance in the domain of live training simulations. This is particularly evident in OneTESS' terrain simulation capabilities. The OneTESS terrain services solution requires terrain fidelity beyond traditional tactical or simulation capabilities while executing on hardware with the computational power comparable to that of a common cellular phone. This prompted the development of the Live Terrain Format (LTF) capability as a prototype of run time services for the Live domain community. Having successfully established the LTF capability, work continues toward evolving the live terrain services to align with virtual and constructive terrain services. The result will allow building a run time terrain services capability that is truly cross domain. This paper presents how LTF prototypes this evolution with an emphasis on how LTF can evolve in general to meet virtual and constructive requirements, the reasons for doing so, the considerations exploring such a capability, and the long term mechanisms to move forward.

This paper begins with a review of the driving requirements, objectives, and design artifacts of the live domain terrain that drove the development of the OneTESS terrain solution. It then presents findings from the evaluation and application of the live domain terrain services in LTF to the constructive and virtual domains. These findings include interoperability and fair fight considerations along with mitigation strategies for existing technologies such as One Semi-Automated Forces (OneSAF) and the Common Training Instrumentation Architecture (CTIA). Additionally, the paper documents an analysis of data acquisition strategies and possible mechanisms for meeting cross domain data needs. Finally, this paper introduces the Common Live, Virtual, and Constructive (LVC) Terrain Evolution work under the Defense Modeling and Simulation Coordination Office. This introduction will cover the genesis, objectives, tasks, and products of the work. The paper concludes by describing how the effort is enabling terrain services that cross the live, virtual, constructive, and operational domains.

ABOUT THE AUTHORS

Jesse Campos is a Principal Engineer with Applied Research Associates with 13 years experience in software development. He has been involved in the Modeling and Simulation Industry for over 11 years with an emphasis in the area of synthetic natural environments in the Live, Virtual, and Constructive domains. He is currently the OneTESS terrain lead. Mr. Campos holds a Bachelor of Science degree in Electrical Engineering, a Bachelor of Arts in Political Science, and a Masters of Business Administration from the University of Central Florida

Steven Borkman is a Senior Scientist at ARA with 8 years of experience developing synthetic environment software for the simulation community. Mr. Borkman is currently a lead developer of the OneTESS LTF project and Principal Investigator for the OneSAF GPU Integration project. Mr. Borkman holds a Bachelor of Science degree in Computer Science from the University of Central Florida.

Gregory Peele is a Staff Scientist at ARA with 3 years of experience developing synthetic environment software for the simulation community. Mr. Peele is currently a lead developer on the OneTESS LTF project. Mr. Peele holds Bachelor of Science degrees in Computer Science and Mathematics from the University of Central Florida

Chuck Campbell is a Principal Computer Scientist at ARA. He has 18 years experience developing Semi-Automated Forces (SAF) software. He has been a developer and technical lead for Synthetic Natural Environment programs including CCTT, WARSIM, OneSAF, and the Rapid Unified Generation of Urban Databases (RUGUD) database generation effort. Mr. Campbell holds a Bachelor of Science degree in Computer Science from Indiana University and a Master of Science degree in Computer Science from the University of Central Florida.

Towards Cross Domain Terrain Services

Jesse Campos, Steven Borkman, Gregory Peele, Chuck Campbell

Applied Research Associates

Orlando, FL

jcampos@ara.com, sborkman@ara.com, gpeele@ara.com, ccampbell@ara.com

BACKGROUND

Current run time terrain databases don't support fidelity requirements across the complete Live, Virtual, and Constructive (LVC) training domains. Each domain has separate terrain database formats meeting a subset of the full LVC requirements. Multiple formats and generation processes require additional maintenance and support costs as opposed to having a common terrain database that provides Modeling & Simulation (M&S) capabilities across the full spectrum of activities and operations.

Most of the common terrain formats were developed for the virtual and constructive domains. They were designed to use lower resolution terrain models because their correlation with the real world was not as paramount as with a live trainer and the source was just not available. Since entities in the constructive domain and automated forces in virtual simulations are computer controlled, they rely on artificial intelligence to make decisions. The synthetic environment must provide the pertinent data, such as soil types and route networks, to support the decision making process. These systems must also provide advanced terrain reasoning algorithms to interpret the data. Human players in live simulations use their own senses to infer this information. While this allows the environment to prioritize its effort on line of sight (LOS), it also demands resolution and reality not met by any current systems.

As a result, the respective LVC terrain elements need to be evolved into a broader M&S capability able to support Department of Defense (DoD) wide operational and support objectives. This can best be accomplished through integration of live terrain capabilities into proposed LVC terrain capabilities and thus providing a common terrain capability. The starting position of this endeavor is based providing a prototype common terrain capability by evolving the OneSAF Environment Runtime Component (ERC) to make it more suitable to accomplish the requirements of a single LVC terrain solution. This evolution should involve incorporating the capabilities of the Live Terrain Format (LTF) in a manner which enhances and provides interoperability. This capability can then be evaluated against cross domain functional, performance, and fidelity requirements to reach a final and complete LVC solution. In order to properly understand the effort involved in a complete LVC solution we begin with a review of the LTF as the first live based terrain services.

LIVE TERRAIN FORMAT REVIEW

Live trainers require synthetic environments that correlate with the real world. In the past, this was not possible. The computational power needed to conduct reasoning algorithms on such high fidelity data was not available. Further complicating the issue was the fact that terrain capturing techniques were not capable of delivering source data with high enough resolution to match the real world. This is not the case any more and technology now exists to meet these requirements.

With the advent of satellite imagery, Light Detection and Ranging (LIDAR) capturing techniques, and multi-core powerful computers, we now have the capability to work with such high resolution data. Live trainers requiring higher resolution terrain data can not meet their functional requirements using virtual/constructive, synthetic, natural environments. Live trainers need terrain modeling solutions developed with Live requirements in mind, such as the LTF.

Requirements

Live requirements differ greatly from those of virtual and constructive systems which depend on a strong feature and attribution layer needed by simulation entities for reasoning functions. Live trainers have a human “in the loop.” Intelligent decisions are made by the trainee, not a simulated entity. The trainee uses their senses to make decisions, based on the world around them.

Another major issue for Live trainers is that the training system needs to be portable and carried by the trainee. Developers of a Live training system generally cannot assume that they will have extraordinary processing power available. In reality they tend to have small, portable computers that are fractionally as powerful as typical desktop units.

The LTF was designed specifically for Live trainers who need to work on extremely high resolution data, use limited resources, and function in real time. The following are high level requirements for the LTF:

- to support geometric paring functionality, the data must be on the order of 0.1 meter accuracy in x, y, and z for terrain elevation and features
- to meet the 0.1 meter accuracy, the SNE must support variable grid resolution down to sub meter with an objective resolution of 1 meter,
- the portability requirement necessitates a compact terrain representation to support large exercise areas while using as little data storage as possible (several gigabytes),
- the portability requirement restricts the memory footprint available to 1/16th of standard memory allocations for virtual and constructive SNE solutions, and
- the portability reduces both the available computational power of the CPU as well as the cycles available due to battery life requirements.

In addition, Live trainers must manage real changes in the environment as part of the training exercise. Changes can occur to both the feature elements of the world – buildings can be destroyed – as well as to the terrain itself – craters, ditches, etc. The training system must provide for runtime environment modifications to meet these needs.

LTF Design Features

LTF was designed to meet the requirements of the Live trainer. The major points of focus were:

- Extremely small footprint to be useful with limited resources
- Accurate representation of terrain with high resolution data
- Real time algorithmic support
- Layered architecture which allows composability and best fit solution
- Dynamic environment support

Layered/Scalable Solution

Many of the current synthetic environments store all of the terrain data together in one encompassing terrain data format. When executing terrain reasoning methods for the environment, all of the data is processed together. For instance, in OneSAF, the terrain features are integrated into the terrain surface. In this model, the terrain triangles and features are stored together and related. When a line of sight query occurs, a single algorithm processes both the terrain traversal and feature intersection check.

This "one size fits all" solution prohibits creating algorithms optimized specifically for a certain data type. In LTF, disparate data types are separated into their own storage and functional layers. This approach allows the development of specialized algorithms for each data type.

The layered architecture allows for a scalable and composable system. The architecture enables users to compose their own system. The layered format will allow the trainee to configure their system to execute without unneeded layers. For example a routing layer, used by constructive entities to determining the best route to travel on may be useless to a live trainee. Omitting the unneeded road network layer reduces the storage required on the resource-limited player unit.

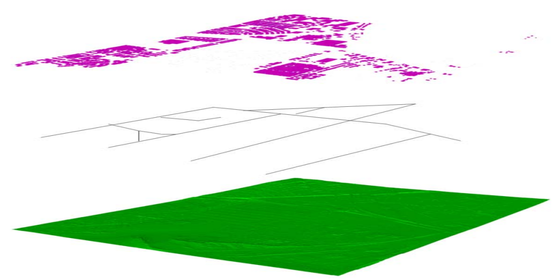

Figure 1. Different layers within a single LTF demonstrating each layer's independence

Storage Layers

The initial release of LTF was composed of two layers: terrain skin and volumetric features. The following is a brief overview of the two layers with additional information found in Reference 1.

Terrain Elevation Layer

To meet database size and performance requirements, the surface elevation is stored in regular grids. The terrain is divided into small 1-km^2 terrain pages using a local tangent plane. The terrain surface is stored in a one-meter spaced grid, with each post representing the terrain surface height in decimeters as a 16-bit integer. The elevation grid for a single page only occupies about 2 MB of memory.

In order to optimize the line-of-sight processing, the terrain is stored in a hierarchical tree. The terrain tree is composed of culling grids which contain the top level elevation of a 10 post by 10 post area, where a *post* refers to a single elevation value. The LTF terrain skin, in its default configuration, is represented by a three-level tree. The bottom level providing the highest resolution is the 1-meter spaced grid, the middle grid is a 100x100 cell culling grid, and the top level is a 10x10 cell culling grid. The number of culling grids and the post spacing of the grids are configurable by the user. The culling grids add minimal storage overhead to the tree structure and the performance gain well justifies the slight increase in storage.

Volumetric Feature Representation

Volumetric features include any objects other than the terrain that are capable of blocking line-of-sight. They are typically small in spatial size and quite numerous, which creates a significant challenge in using them efficiently. Examples of common features include trees, buildings, and light poles.

LTF represents features as leaf nodes in a bounding-volume hierarchy (BVH) tree, which is a spatial tree commonly used by graphics and gaming applications for ray-tracing and collision detection. Each node in a BVH tree is a spatial volume that fully contains all of the volumes of its child nodes. Intermediate “culling” nodes organize features that are spatially close to each other so that a quick intersection check on the culling node can potentially eliminate a large number of feature nodes from consideration in the line-of-sight algorithm. For line-of-sight calculations, BVH trees are somewhat slower than spatial-partition trees such as *k*D-trees, but were chosen for their significantly better update performance to support dynamic terrain events.
The LTF BVH tree supports using arbitrary geometry types for both intermediate “culling” nodes and feature geometry nodes through a common geometry interface

Like other database formats, LTF supports the concept of attribution for features: any feature in the BVH tree can have a set of associated attributes. The only attribute currently defined for OneTESS is the “Material” attribute, which specifies the material type for the feature. Many properties that are commonly represented as attributes in other formats, such as feature length, width, and height, are instead explicitly represented by the feature's geometry in LTF.

Dynamic Terrain Representation

One meter post spaced grids do not provide the needed resolution to properly integrate dynamic events such as foxholes and craters. A foxhole can be dug into the terrain in any location, not just on post boundaries. For Live training, correlation of dynamic events with the real world is paramount. The terrain grid is designed to allow higher resolution grids in small areas. However, due to the size constraints of the player unit, this currently is not feasible. Therefore, the current dynamic terrain functionality is modeled with terrain features only.

The BVH structure is designed for use with dynamic changes. The performance cost to create, modify, or delete a feature are less for a BVH than with other tree structures. When a new feature is created, the correct bounding volume is identified and the feature is inserted. If needed, the containing volume will either have to be modified or created. If a feature is deleted or modified, very minimal changes need to occur.

Since the terrain skin cannot accommodate dynamic deformations, OneTESS will use features to model all dynamic events that alter the terrain surface. *Subtractive features* are a special abstract feature type representing a volume without accurate geometry. Future goals are to alter the actual terrain surface, eliminating the need for subtractive features.

CONSTRUCTIVE AND VIRTUAL APPLICATION

LTF was designed to meet the needs of live trainers. A considerable amount of adaption needs to occur to LTF to meet the needs of virtual and constructive systems. Live requirements focus on the physical, three-dimensional world. The terrain objects capable of stopping line-of-sight or affecting a bullet’s flight. Virtual and constructive terrains need a wide array of terrain objects and attribution for use by simulated entities for automated planning and reasoning. To further complicate the issue, virtual worlds need the information to be graphically created and to look realistic. This could include information such as textures, colors, etc. As SNE solutions attempt to optimize for specific domain interoperability and fair fight issues start to surface. As a result, this section will conclude with discussion and specific strategies in this area.

Constructive Domain

Pinpointing the functional requirements of the constructive domain is a difficult process. Constructive simulators encompass all systems that contain computer controlled forces. In order for simulation entities to have realistic behavior the environment needs to provide ample amount of information on which to reason. Physical obstructions provide insufficient information, and attributes like soil type, foliage density, and snow depth may be needed.

The environment also needs to provide terrain reasoning algorithms for the simulated entities. Most of the time, with live and virtual trainers, the route to take between two locations is decided by the user. With simulated entities, the proper route to use has to be determined algorithmically by traversing a routing network. There may be several routes between two points, and the system has to be able to determine the proper route.

LTF was designed to meet the needs of live trainers. It is designed to provide the environment data essential to compute line of sight and, because of the limits imposed by portability requirements in the live domain, does not contain the breadth of environment data vital for constructive systems. However, the core design principles make LTF well suited to meet the needs of the constructive world.

LTF leverages a layered architecture to allow for an ideal solution for each particular problem and data type. For example, the terrain, features, and attribution are all stored in separate layers. This allows for an ideal set of data structures and algorithms designed for the specific terrain element type.

To meet the needs of the constructive domain, LTF can simply extend the feature layers to provide the necessary data. For instance, a road network layer can be added to meet the needs for routing. The feature attribution layer can be enhanced with a more robust data model to provide the needed attribution for simulated entities to be reasoned on.

The layered architecture also allows for a composable synthetic environment. Different applications can configure the LTF with the layers need to satisfy their particular requirements. We will use the above feature attribution example to further illustrate the benefits of a composable system. A live trainer with limited resources has little need for a detailed feature attribution layer. Such a system can configure the LTF with a limited attribution layer, allowing a smaller data footprint on the embedded equipment, but still containing all of the required content.

A semi-automated forces (SAF) system, which simulates computer controlled entities, may need more robust attribution to meet the needs of the artificial intelligence algorithms. The SAF, running on a desktop system with ample resources, can instantiate the LTF with an expanded attribution layer. The other LTF layers are identical on both systems.

LTF will require content and algorithm extensions to meet the needs of the constructive community. However, many of the environment needs are shared between the live and constructive domains. Examples of the overlapping data needs include terrain elevation data, three-dimensional volumetric features, feature attribution, and line of sight processing. Live trainers have very high resolution requirements compared to constructive trainers on these shared needs. Initial results indicate that LTF is very capable of meeting the performance and resolution requirements of constructive trainers. It also meets these needs on the limited resources of the embedded player unit.

Virtual Domain

The virtual domain involves real people operating simulated systems. Virtual simulations inject a human-in-the-loop in a major role by exercising skills such as driving a tank, acquiring targets, or firing a system. In this domain, the natural environment representation includes external features such as man-made structures and systems with which trainees interact. The virtual environment typically requires consistency over correctness. The data must encompass the terrain relief, terrain imagery, and 3D models of natural features, man-made cultural features, and military vehicles and equipment. In addition, virtual trainers can require specialized content, such as texture mapping functions, levels of detail, display priority schemes, light and shadowing parameters, etc.

Virtual and constructive trainers can achieve terrain interoperability by modifying real world data to fit the needs of the system. The live domain does not allow that simplification and poses greater problems such as how live and virtual entities are to interact. Specifically, while live entities can be relatively easily injected into virtual systems, it is not currently feasible to inject virtual entities into the live domain.

As a result, current and near future training solutions will need to work around these limitations and provide interactions if not necessarily interoperability. To this end, design features exploited in LTF for the constructive domain are also applicable to the virtual domain. The layered architecture allows storage of visual data such as polygons and textures in distinct separate layers, allowing use of virtual data structures and algorithms. Additionally, enhancements to the attribution layer can provide the needed attribution for human interactions. Finally, as the virtual domain progresses further toward dynamic generation or run time publishing, their trainers may opt to leverage the correlation of the LTF with the real world source data for mission planning or greater accuracy in general exercises.

Interoperability and Fair Fight

Interoperability refers to a set of diverse systems working together. The military has a wide range of computer-based training systems that teach most aspects of military operations. From vehicle repair to logistics to tactical planning, computer-based simulations can effectively train many facets of processes and procedures in a cost-effective manner. While these trainers are effective in and of themselves, interoperability attained through linking these different simulators together can train a wider range of skills. Train as you fight.

Fair fight with respect to training simulator interoperability refers to making sure one simulator does not have an unfair advantage over another. Fair fight issues can stem from a number of sources, but typically trace back to differences in data and algorithm fidelities. For example, different line of sight algorithms may return different results, allowing one entity to "see" another while the "seen" entity believes it is hidden. Fair fight issues can result in frustration and mistrust in the trainer.

Terrain correlation is typically the most obvious sign of fair fight issues when interoperating. Historically, each program defines its own terrain representation based on its requirements. This includes the terrain features, attribution, and resolution. These characteristics often drive the selection of source data from which to build the terrain and the tools used in the generation process. Subsequently, terrain databases representing identical geographic areas can vary widely from simulator to simulator. If the data on which simulators are interoperating does not correlate, there is little hope of fair fight.

Similarly, fair fight issues can manifest within terrain query and reasoning algorithms. The calculation of line of sight given above is a good example. Even if the simulators operate on identical terrain, different algorithms can produce different results. Differences reported at the terrain level can ripple up through entity and behavior models. One simulator may see collisions where none should occur, damage from detonations that seem out of range, odd driving behavior, and other anomalies.

Fair fight and interoperability issues extend well beyond the terrain model. However, due to their reliance on the terrain, higher functions have no chance of achieving fair fight without correlation at the terrain level. Let's consider two technologies and paths possible from the terrain perspective.

Mitigation for Existing Technologies

The first technology is OneSAF the Army's premier Computer Generated Forces/Semi-Automated Forces (CGF/SAF) system which operates on the OneSAF Environment Runtime Component (ERC). As mentioned previously to support the virtual and constructive domains, the data required can be low resolution in the area of 100 m and 60 m terrain polygons. Since the simulated entities, or users training on virtual system, are completely immersed in the virtual world, the terrain resolution is acceptable. Even though it is not as detailed as reality, it may very well be good enough to accomplish the training. Live does not have this benefit as Figure 2 exemplifies.

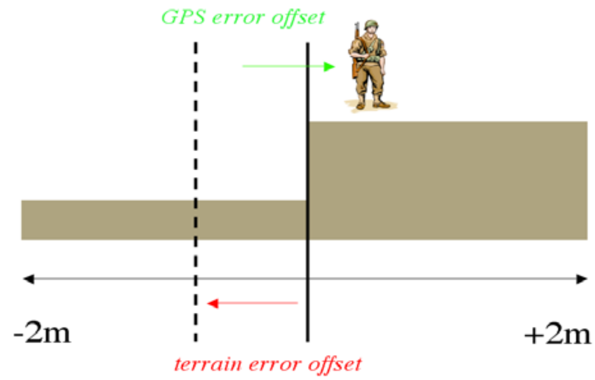

Figure 2. Foxhole example highlighting high resolution data requirements in Live training

In the live domain, the training system must make an adjudication based on whether the soldier is in the foxhole or not. While the GPS error comes into play, the terrain error can overwhelm the calculations. Specifically, if the data is not accurate within the GPS error, then the system will not be able to determine the soldier is in fact protected within a foxhole. Additionally, if the resolution is too low, then a foxhole the size of 1m by 2m will not be captured. In a constructive world, the AI determines that the entity goes into the foxhole, no matter where the foxhole is on the SNE terrain. In the virtual domain, the entity will maneuver into the foxhole based on the terrain displayed, so again it does not matter if the foxhole is accurate.

Consider another problem with inaccuracies dealt with on the live domain related to elevations termed overshoot and undershoot, illustrated in Figure 3. Due to GPS vertical inaccuracy, we assume the use of 2-D GPS position to locate the player in the terrain map and then using the recorded elevation to infer elevation. If the terrain is, for example, actually 1 meter lower near the detonation point of the indirect fire round than the terrain database shows, there is an undershoot situation as shown below. Similarly, if it is 1 meter higher, then there is overshoot. Overshoot can be calculated from the elevation angle as $O = E / \tan \prod$, where $\prod$ is the elevation angle of the shot, E is the elevation error (difference between the database-elevations of shooter and target minus the difference between actual elevations of shooter and target). Thus, for shallow angle shots, mandatory for M203 and Mk-19 grenade launchers if used within doctrinal ranges, there is a large multiplying factor. For example, one meter of elevation error translates to approximately 24 meters of overshoot for the M203 used at a range of 50 meters. (Note: This is one of the reasons the terrain accuracy requirements specify 0.1 meters) Again, the virtual and constructive domains do not encounter this situation because there are no elevation errors; the synthetic terrain is the de facto terrain.

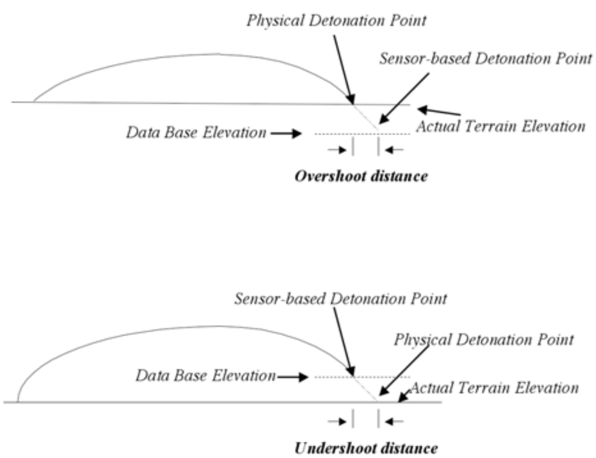

Figure 3. Overshoot example demonstrating error sensitivity in Live training

Given these discrepancies, there are several options to handle interoperability. The base solution is to prohibit these systems from interacting. This may sound tongue in cheek, but in actuality, there are usually some minimum requirements for systems to operate, e.g. protocol compatibility. Additionally, all systems can operate at the lowest common denominator. In such a paradigm, each system would reason and operate on the data of the weakest component. This would require the live domain to use terrain data that would make its requirements unreachable. As a result of this paradigm, in order to interact with each other, training is lessened. A final paradigm presented provides one central mechanism from which every system determines its own results and behaves accordingly. This is similar to the LVC adapter used on OneSAF, where OneSAF calculates results for constructive entities and live systems adjudicate their results on their side. Since there is a relatively clean interface, this will allow interaction if not interoperability. Unfortunately, the terrain does not provide a clean interface. Specifically, when two different terrain services provide two different answers to the same question because of data differences, interoperability and fair fight is severely hampered.

A second technology to look into for mitigation of interoperability deals with possible solutions terrain services in CTIA. Consider the following situation:

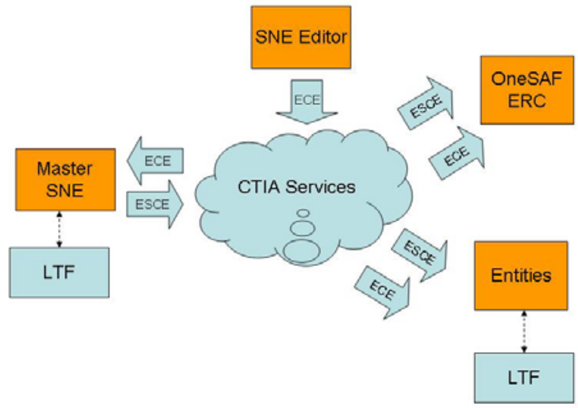

Figure 4. Example of CTIA Services used by multiple terrain service providers and ensuing interoperability problems

Figure 4, illustrates the situation in which a live domain solution based on LTF is interacting with a constructive terrain using OneSAF ERC. CTIA Services is the software that provides the messaging and LT2 components required for the live domain. Among those components is a SNE Editor which allows modification of the terrain that has occurred in the real world, be that building changes, foxholes, berms, etc. This information must be conveyed to a Master SNE component which contains a picture of the real world. Additionally, all of the live entities must also receive this information as they require their own LTF data in order to meet live training requirements. Finally, the OneSAF ERC which provides the terrain services for the constructive entities must receive the information. The above case depicted is an addition to the terrain. When the addition is made through the SNE Editor, a CTIA EntityCreatedEvent (ECE) is sent out to all who are registered to received this message, in this case the Master SNE, OneSAF ERC, & the Entities component. The Master SNE which maintains the ground truth LTF receives the ECE message and must process it to provide final LTF information such as the unique feature identifier required and then transmits this information out through an EntityStateChangedEvent (ESCE). The interested parties being the Entities component and the OneSAF ERC. The modification and deletion of a terrain feature is similar to the above interactions.

There are some fundamental issues that must be addressed for interoperability to occur in the above situation. Specifically, the OneSAF ERC must take into account the changes made by the SNE Editor and the Master SNE in order to maintain correlation. In the case of CTIA, this cannot be done through software since the architecture is based on independent components. Additionally, the issue of how terrain modifications can occur must be determined. Specifically, there could be other SNE Editor like components which modify the terrain and they must follow the same process as the SNE Editor or interoperability. As a result while the OneSAF LVC mitigation strategy revolved around software, in this case the mitigation strategy revolves around a defined policy. So in this situation, the behavior is understood and documented so that interoperability is not destroyed.

DATA ACQUISITION STRATEGY

In order to properly train across all of the LVC domains, a concerted and structured approach must be applied in order to capture and acquire the data necessary for training purpose. A training system without the prerequisite data necessary for optimal performance is an incomplete system. As a result we now turn our attention to the data acquisition strategy across with an emphasis on the live domain aspects.

Data Acquisition

The training community is very much at the mercy of the operational community to produce training databases. This has historically meant that training solutions take raw data from sources such as the National Geospatial-Intelligence Agency (NGA) and then process them over months, weeks, and days for a useful product. While the time required to process such databases would theoretically continue to decrease, the savings is offset by ever increasing functional requirements on the part of training systems. Thus, a cycle exists in which training systems become more complex which in turn require more data which requires more processing which finally results in training data that is considerable different than the raw source data that is collected. The live domain brings in a new element to this cycle which can significantly affect the status quo of the terrain training database production cottage industry. Two specific elements on which we will focus is the use of high resolution data in the terrain database generation process and how a live solution like LTF makes just-in-time data fusion possible.

High Resolution Terrain Data Usage

Most available high-resolution data is collected using aerial LIDAR sensors. These sensors can collect 1m and sub-meter data within the specified 10 cm vertical accuracy required for live training. Most collections are over somewhat small geographic areas, tending toward tens of kilometers per side. This is understandable given collection costs, size of training units requiring such high-resolution data, and current trainer ability to represent data at this level of detail.

Collected LIDAR data is captured as a non-regularly spaced point cloud, due to movements in the aircraft during collection. The collected data is processed to extract bare earth terrain and terrain features such as buildings and trees. The resolution of the collected data affects the types of features extracted, as features near or below the elevation sample rate are lost in the noise. The processed data results in a Digital Elevation Map (DEM) describing the terrain surface and a set of vector files describing the terrain features. The DEM and vector files are inputs to the terrain generation process that creates the run-time application databases.

Hyperspectral imagery can augment LIDAR collection with surface material types. Hyperspectral imagery processing for a wide variety of soil, vegetation, and construction materials remains a largely manual, time-consuming task. Imagery classification is dependant on climate and time of day, so the classification process must account for atmospheric conditions. Lacking this process, there is no effective method for detailed terrain attribution beyond geometry. One could make the case that the extra expense of collecting hyperspectral imagery along side LIDAR is justified in anticipation of emerging techniques to automate the classification process. Otherwise, the training community will have accurate geometry representation with inferred material attribution.

Efforts are underway to collect vehicle-based LIDAR data. Sensors are able to capture very accurate (1cm and below) data for building fronts at the ground level. So far, this results in very good 3D views of the area driven. However, the amount of data collected is enormous and the ability to process this data into full 3-dimesional geometries for use in semi-automated forces trainers is immature at best. That is not to say ground-based LIDAR does not have a place in mission rehearsal and planning, but tools and techniques are years away from practical incorporation into simulation training terrain databases.

Ground-based LIDAR collected data is now beginning to make its way into the terrain database generation, but it is hampered by a couple of issues. First, the bulk of its size limited its usability on current systems. Additionally, the lack of high resolution requirements allowed for smaller, less detailed data usage. Finally, the mandate to cut processing time from months, to weeks, to days, to hours, reduces the viability of using this data for processing. These issues are not as relevant in the live domain as well as the embedded training environment which is a big element in the future of military training.

Just In Time Data Fusion

Historically, the warfighter trained with different equipment than what was used in operations. This produced physically different equipment that provided operationally equivalent training. This was especially the case for the virtual domain as seem by such capabilities as the Combined Arms Tactical Trainer (CATT) family of systems as well as the constructive domain. In the live domain, a driving requirement is to train as you fight which has reduced the amount of training equipment. The live domain embraced the concept of embedded training which embedded training capabilities into operational systems. This is consistent with the Operational Test Command which instruments operational systems.

With the advent of the Future Combat System, the distinction between training equipment and operational equipment, essentially was removed. The warfighter would now use the same equipment in both training and operational situations. In essence, it has moved from “embedded” training to “organic” training. While this has brought out many issues and obstacles, it also represents a significant opportunity. An opportunity to provide synergies among the training and operational communities that the entire modeling & simulation industry can benefit.

Consider an organic training based system in which the terrain information is provided as operational data due to size limitations and to facilitate data fusion as illustrated in Figure 5.

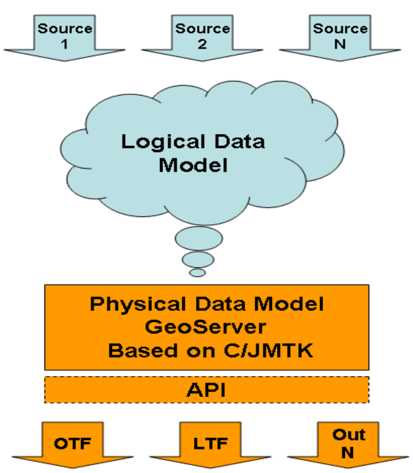

Figure 5. Conceptual diagram of data transfer process from source data to actual runtime terrain services

Designing the system with two different terrain paths not only creates significant storage requirements, but also creates multiple paths that result in inefficient data fusion complications. An implementation of a terrain server for a cross domain organic training capability such as the Future Combat System requires data sources from NGA, unmanned vehicles, sensors, field reports, etc. These sources are processed and fused into a logical data model with an implementation into a physical data model such a GeoServer based on the Commercial Joint Mapping Tool Kit (C/JMTK). In this paradigm, the data must now be processed and provided through an API for use in training capabilities across the LVC domains. The optimal solution is that the GeoServer contains all the data as close to raw source data as possible and then it provides the data through the API and converts it following the Just In Time (streaming source data) paradigm per Figure 6.

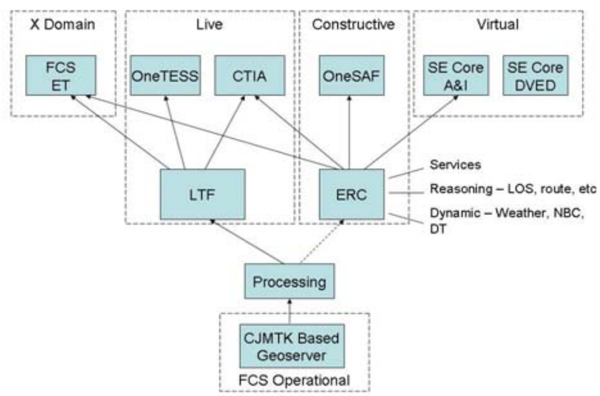

Figure 6. FCS architectural diagram demonstrating cross domain component usage and streaming

This paradigm puts a premium on run time data consistency with source data in structure. The closer the run time data is to the GeoServer data, the more viable streaming source data becomes. As a result, different run time formats will have varying degrees of success under this new mantra.

In the case of the live domain, the LTF is ideally suited for streaming source data compilation. This is due to its nature using a layered based approach which maximizes real world correlation and source data consistency. Currently, the LTF still follows the paradigm of pre runtime database generation, but its process can be converted to an in memory process to provide live requirements by first creating a concurrent generation system as illustrated in Figure 7.

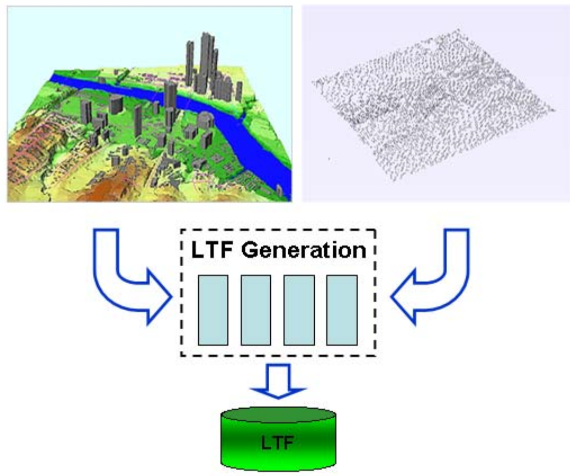

Figure 7. Conceptual diagram showing how the independent layers of source data can be processed separately and in parallel to produce an LTF

Currently the process uses the source data provided as elevation and features and sequentially produces an LTF. In order to compile this information in memory, the process needs to be made concurrent and take advantage of the LTF design capabilities built in for concurrent programming. The generation can be separated into distinct concurrent processes which expedite the processing.

Once the generation is made concurrent, the entire process is converted into an in memory process by modifying the LTF terrain services to read directly from the C/JMTK API and populate the in memory structures of the LTF library. This is possible because of the severe memory constraints levied by the portability requirements in the live domain. This requirement forces the LTF and live terrain services to provide strict memory control and independent service components. For example, LTF provides a LOS capability for features and one for elevation that are independent.

COMMON LVC TERRAIN EVOLUTION

This leads us to the final topic in this paper, work begun for the Defense Modeling and Simulation Coordination Office (MSCO) to begin the process of building a run time common LVC terrain capability.

Genesis

Current terrain databases do not adequately support fidelity requirements across the LVC training domains. Domain focused terrain databases that meet subsets of the full LVC set of requirements are the norms. This limitation has resulted in the development and support of many run time terrain databases, as opposed to one common terrain representation that provides M&S capabilities across the full spectrum of activities and operations. For example, in the constructive domain, modern SAF systems support a wide range of services and functionality. And in some cases (e.g. CCTT) the same terrain services must support applications beyond SAF (manned simulators, user workstations, etc.). The need to support services as diverse as height of terrain, line of sight, route planning, collision detection/avoidance, and cover/concealment complicates trade offs, quite often to the extent that multiple terrain formats are created within the context of a single system. ModSAF/OneSAF Test Bed (OTB), CCTT SAF, and OneSAF all use multiple on-disk formats to handle specialized functions (e.g. OneSAF stores route planning networks separate from the file containing terrain polygons).

This situation has led to formats established in one domain being extended to other domains without properly considering specific optimizations involved in each domain. The terrain requirements for live training systems signify a complete paradigm shift from the virtual and constructive systems. High resolution data is no longer a luxury but a necessity. Live trainers blur the line between simulation and reality in such a way that real world and virtual world correlation are paramount. To compound the problem, real-time terrain reasoning functionality with limited resources is essential. For the most part, live training systems have to execute on more complex, higher resolution data in a smaller performance allocation than their virtual and constructive counterparts. Whereas live trainers have more stringent requirements for line-of-sight (LOS) algorithms, they have little need for terrain reasoning services as compared to constructive formats. Development of a higher level system interface can allow these formats to coexist and complement each other to support the full spectrum of LVC terrain requirements

Objectives

The LVC terrain evolution promotes interoperability by providing a runtime terrain services and associated tools and data to enhance a fair fight through a common M&S capability. Starting with a known M&S capability across domains allows meeting the goals of supporting a wide range of DoD interests, providing timely and credible results, and reusing investment across communities.

The training community benefits from a mechanism to inject live training to enhance realism while still maintaining interoperability. Training also benefits from a standard runtime format for M&S data along with a common and persistent interface. Additionally, a common high fidelity solution provides the necessary accuracy and realism required for other DoD uses such as testing. Finally, the test and experimentation community will benefit from the significant investments in M&S capabilities that will be leveraged.

The overall objective is to evolve the respective terrain elements in the LVC into a broader M&S capability able to support operational and support objectives across the war fighters need. This overall objective decomposes into distinct goals as follows:

- clearly define standard terrain services interfaces,
- analyze and document tradeoffs across the LVC domains for use in developing DoD wide solutions,
- provide an implementation for expanded LVC software solutions, and
- evaluate and test existing solutions and technologies.

Tasks & Products

In order to meet the objectives, the LTF can be merged with the Objective Terrain Format (OTF) and the capability evolved to support the full spectrum of run time terrain services and support high fidelity representations derived from high resolution data including sub meter LIDAR.
Four concrete products are available for this work. First, a Key LVC Terrain Interface Support Study will be produced. This study will identify the key elements of the LVC Terrain Interface to be supported.

The second product is a LVC Terrain Interface Specification that defines the full range of runtime terrain services. This will support multi-resolution, composable, and run time configuration capabilities. An initial version of this deliverable will be provided by December 2008 for community review and input. The final version is scheduled for June 2009.

The third product is LTF/OTF based OneSAF ERC. This will provide an ERC capability that seamlessly supports services required by live, virtual, and constructive domains by building on LTF and OTF services, for the key elements specified in the Key LVC Terrain Interface Support Study. This product is expected to be completed by June 2009.

The final product is High Level Design for LVC Support. This produce will provide an initial design of the ERC supporting the LVC Terrain Interface Specification. This product will use the design lessons learned from the previous products to define an initial design that can be built upon in future work.

TAKEAWAY

Evolving domain specific technologies into capabilities that support the full LVC domains provides benefits throughout the community and enhances the warfighter training. The LTF provides initial capabilities in the live domain which meets live training requirements. Leveraging this capability into existing capabilities such as the OneSAF ERC will allow the development of a truly cross domain LVC terrain services.

References

S. Borkman, G. Peele, & C. Campbell, *An Optimized Synthetic Environment Representation Developed for OneTESS Live Training*, I/ITSEC 2007, Orlando, FL

R. Hall & M. Janisz, *Sensor Error Tradeoffs for Geopairing*, OneTESS Modeling and Simulation Team, June 8, 2006.

Bresenham, J, *Algorithm for Computer Control of a Digital Plotter*, IBM Systems Journal, 4(1):25-30, 1965

Lauterbach, Yoon, Tuft, Manocha, *RT-DEFORM: Interactive Ray Tracing of Dynamic Scenes using BVHs*, In Proc. 2006 IEEE Symposium on Interactive Ray Tracing, 2006

Hall, R.J. & Chludzinski, J (2007) *Timely Provision of Terrain Elevation Data for Geometric Pairing Computations*. I/ITSEC, 2007

OneTESS Program, *Synthetic Natural Environment OOS Reuse Report*, 29 December 2006

Baer, Campbell, Campos, & Powell, *Modeling Terrain for Geo-pairing and Casualty Assessment in OneTESS*, Defense + Security 2008