DEVELOPMENT OF FUTURE COMBAT SYSTEM SYNTHETIC NATURAL ENVIRONMENTS

Benito Graniela, Daryl Siddon
SAIC ASSET Group
Orlando, FL
Benito.Graniela@saic.com, Daryl.Siddon@saic.com

ABSTRACT

The United States' Future Combat Systems (FCS) includes a wide variety of military platforms, which will help soldiers in conflicts of the 21^{st} century, including: manned and unmanned, ground and air, sensors and weapons capabilities. FCS includes the Objective Force Warrior (OFW) combatants and various unmanned sensors, which operate within close-in environments requiring a high level of detail in the Synthetic Natural Environment (SNE). This paper describes the development of a set of correlated Semi-Automated Forces (SAF) and visual SNE for FCS experiments. The development and significant features of three SNEs are discussed. The first SNE is an urban database that contains a large number of Multiple Elevation Structure (MES) buildings (with interiors). The challenges associated with this quantity of MESs along with the implemented solution are discussed. The second SNE discussed is a mountain database that contains complex cave structures with different types of ground access, and tunnel geometries. The challenges with the junction of the terrain skin and tunnel access along with various solutions are discussed. The third SNE is a complex and dense jungle database with restricted mobility and line-of-sight. The challenges with modeling a jungle SNE that permits warfare along with the implemented solutions are discussed.

ABOUT THE AUTHORS

Benito Graniela is currently working as a member of the Human Simulation group at SAIC in Orlando, FL. The emphasis of his work has been on building correlated visual and SAF urban SNEs with building interiors (CTDB multi elevation structures) to support SAF behaviors in DISAF. Prior to SAIC he worked at the Institute for Simulation and Training on interoperability of visual distributed SNEs and 3D graphics benchmarks. Mr. Graniela earned a B.S.E in Electrical Engineering from the University of Puerto Rico in 1985 and a M.S.E in Computer Engineering from the University of Central Florida in 1992.

Daryl Siddon is a Program Manager for the SAIC ASSET Group in Orlando, FL. Mr. Siddon received his B.S. in Computer Science from the University of Central Florida in 1991 and has been actively involved in the Modeling and Simulation community since 1992. During the last five years he has been specifically involved with development of Dismounted Infantry simulations.

DEVELOPMENT OF FUTURE COMBAT SYSTEM SYNTHETIC NATURAL ENVIRONMENTS

Benito Graniela, Daryl Siddon
SAIC ASSET Group
Orlando, FL
Benito.Graniela@saic.com, Daryl.Siddon@saic.com

BACKGROUND

The Future Combat System (FCS) Program (Boeing WWW) is defining a set of weapon systems for the United States (US) military, which will allow the US to be more effective in future conflicts. These systems will make heavy use of unmanned systems as well as non line of sight (NLOS) weapons. FCS makes use of simulation to select the systems and components which are the most effective. For this reason FCS needs a set of Synthetic Natural Environments (SNEs) that can be used as test contexts for the systems.

The Dismounted Infantry Semi-Automated Forced (DISAF) Group (SAIC WWW) of Science Applications International Corporation (SAIC) in Orlando, Florida has enhanced DISAF and developed SNEs to support FCS exercises/experiments. These exercises/experiments were conducted by Ft. Benning, Georgia during the 2003 calendar year. The FCS exercises and experiments required new SNEs for simulation evaluation of FCS concepts implemented as DISAF models of systems, vehicles, sensors, weapons, and munitions. This paper concentrates on the descriptions and the development of the SNE.

Three SNEs are desired by Ft. Benning to support the infantry focus of FCS. The first of the environments developed was an "urban" city with a high concentration of buildings in a small area. Another developed environment was a "mountain" region that included underground tunnel structures. Requirements for a third "jungle" SNE are currently being developed. Visual and Semi-Automated Forces (SAF) synthetic environment representations were developed to correlate with existing databases used by the Lawrence Livermore National Laboratory's Joint Conflict and Tactical Simulation (JCATS) application.

Reality By Design's (RBD's) Soldier Visualization Station (SVS) application and SAIC's DISAF application utilized the resulting databases. The SVS database format was OpenFlight (MultiGen WWW) and the DISAF database format was the Compact Terrain DataBase (CTDB). DISAF is a derivative of the OneSAF Testbed Baseline (OTB) that has enhanced dismounted infantry capabilities. For the FCS exercises and experiments, a new set of FCS weapons systems were modeled. Since Dismounted Infantry frequently operate within a close-in environment, buildings with interiors are represented in CTDB as multi-elevation structures (MESs). MESs are composed of enclosures (i.e., rooms) and apertures (i.e., doors and windows). To permit subterranean movement, a tunnel can also be represented as an MES (Graniela 2001a).

DATABASE DEVELOPMENT

Process

Over the past five years, SAIC has developed and refined the techniques for development of high resolution urban environments within OpenFlight and CTDB SNEs. The current process for the development of correlated visual and SAF databases includes the use of TerrEx's "TerraVista" terrain modeling tool, MultiGen's "Creator" 3D modeling tool, and the OTB "recompile" tool. Additionally, SAIC has developed a set of tools for converting OpenFlight building models into MES reader (i.e., ".rdr") correction files. These tools help to achieve the highest degree of correlation

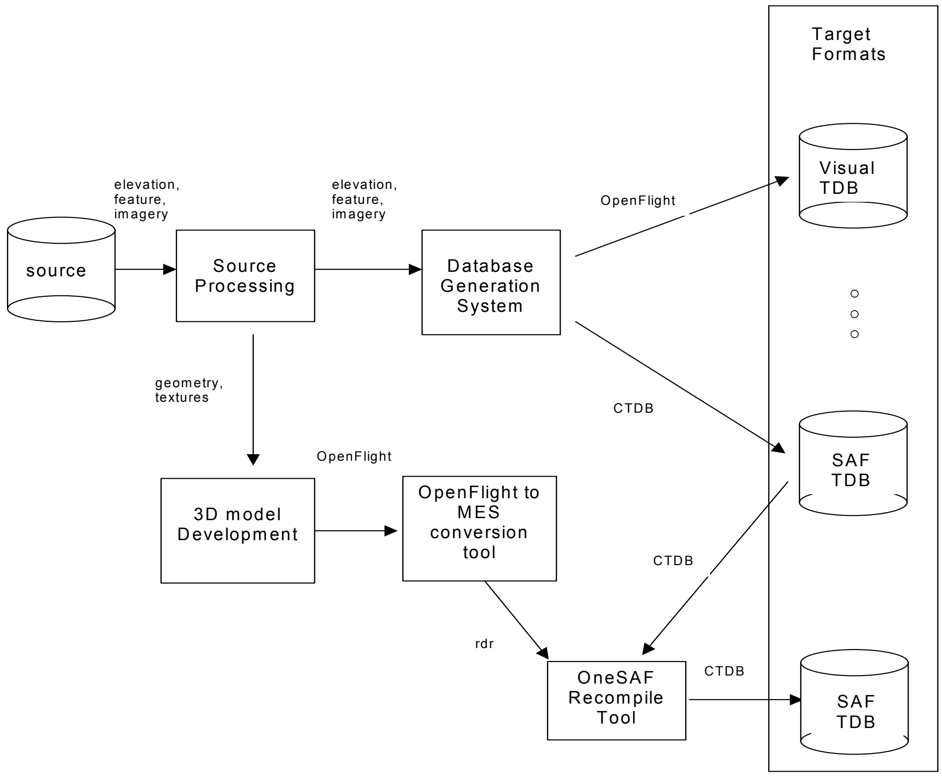

Figure 1 - Database development process.

and rapid generation of complex MES environments.

Figure 1 shows the current development process. The TerraVista (TV) database generation application provides an easy way of importing elevation and feature source data and mapping attributes of the data to computational models called Feature Models (FM). TerraVista processes source data and generates intermediate representations which can then be exported to a number of formats. This is how the initial OpenFlight and CTDB files (containing roads, building volumes, and trees) are generated for a given database. SAIC recently helped test versions of the TerraVista application that can insert MES directly into the CTDB. Since this TerraVista feature is still new and experimental, and our project required a stable development approach, we used our process. The Multigen Creator application was used for the generation of the buildings with interiors. This tool can create 3D models with a specific group and object node hierarchy in the way necessary to generate the MES correction files. These 3D models are converted into correction files for the CTDB using a custom application which extracts the geometry and attributes from OpenFlight. The OTB "recompile" tool is used to merge the correction file into the existing CTDB. Other tools used as part of our development process include ESRI's ArcView Geographic Information System (GIS) application and Adobe's Photoshop graphics application. The ArcView application is used to pre-process elevation and feature source data before it is imported into the TerraVista tool. The Photoshop application is used for the development of visual model textures.

Urban Database

The urban database contains a high density of city-like features which includes a municipal building, apartments, hotels, mosques, theaters, housing, residential houses, high-rise buildings, and warehouses. This SNE was based on the content of an existing JCATS database. The JCATS database itself is based on a Central Asia city. Since the JCATS building representation does not provide 3D visual details, the architectural and texture features were developed based on typical a Baku architecture.

Custom software tools were developed by SAIC for the extraction of information from the JCATS database. The JCATS application can be used to generate a JCATS text format (.daf) file from the corresponding binary version. The text format describes the database features in a human readable format (LLNL 2002).

For our purpose, the JCATS terrain file contains two types of information. The first part contains the terrain feature geometry (or shapes) and the second part contains hierarchical grouping information used by the terrain features. Version 4.0 has nine different sections, with each part grouping similar data. The nine parts are:

1. JCATS File Version
2. Date/Time Stamp
3. Lower Left corner of the terrain file
4. Tile Container
5. Trafficability Info Container
6. Trafficability Name List
7. Terrain Feature Container
8. Terrain Ground Contour Container
9. Terrain Elevation Grid

The "Terrain Feature Container" section includes the terrain features for roads, rivers, buildings, ground polygons, tunnels, and so on. Since MES are the focus of this paper, we will describe the format of the JCATS Enhanced Building Terrain Features (EBTF) in more detail. JCATS EBTF represents buildings with interior features and floors. In essence, it is an enhanced 2D representation of a building. The EBTF components are:

1. *ShellBuildingTerrainFeature* (contains information about the exterior of the building)
2. *NumFloors*
3. *FloorOfBuilding*

The *FloorOfBuilding* feature contains a floor number, height and a feature called *FloorPlanOfFloor* which describes the layout of the rooms in a floor. Each floor plan contains information which describes the interior and exterior walls as linear features. These linear features contain door and window linear features as well.

Roads and building footprints were quickly extracted from the JCATS database and converted into shape files using a custom ArcView script. The shape files were imported into TerraVista to expedite the generation of a "rough cut" database. At this point in the process, the database contains generic TerraVista building's exterior and interior textures and the CTDB contains volume representation for the buildings. This process is typically completed in a few days. The result is shown in Figure 2. The corresponding CTDB database had volume features for all the buildings as well as the roads and individual trees.

Figure 2 - Visual representation of the "rough cut" urban database.

This representation does not include building interiors, which are required for our Dismounted Infantry simulation. Ft. Benning was interested in having buildings with interiors similar to those in the JCATS database, therefore the volumetric representations of CTDB buildings needed to be converted into correlated MES buildings. In addition the default textures used by TerraVista were not typical of the region so textures needed to be developed for the building exteriors and interiors as well.

The JCATS urban database consists of ten unique buildings with multiple instances of each spread throughout the city. To be able to create instances of the new building models it was necessary to convert the JCATS footprint information to point features for each building. The point feature data contains: location, building type, and orientation. This point feature data is used to create instances of the buildings in the database. This information was derived from the original JCATS footprint orientation using an ArcView script.

In order to meet an aggressive experiment schedule in early 2003, two versions of the urban database were developed. The first one contained shell MES buildings, whereas the final one contained complex MES buildings with interiors. In the first version, simple MES buildings were developed for the 10 different buildings. These buildings match the JCATS outlines but contain one enclosure and at least one aperture to access the building interior. The simple MES buildings were modeled using MultiGen Creator. This visual 3D modeling tool was used to developed the models and generate an OpenFlight hierarchy similar to that shown in Figure 3.

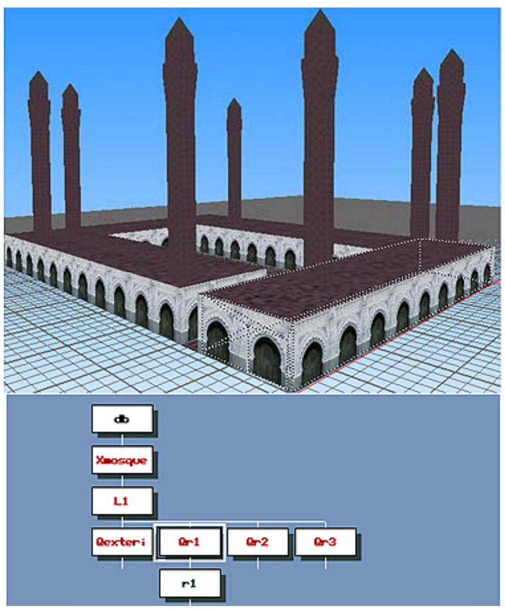

Figure 3 - MES model hierarchy.

OpenFlight defines a node hierarchy that permits organization, attribution, and manipulation of geometry information. In the example, the first grouping node on the hierarchy named “Xmosque” defines the model name, under that the “L1” node groups all level 1 (i.e., floor) enclosures. The nodes “Qexterior”, “Qr1”, “Qr2”, and “Qr3” contain enclosure, aperture geometry, and attributes. “Qexterior” is a special kind of enclosure node and must always be present as it contains all of the geometry which defines the building exterior walls as well as all the apertures that allow access to the building. “Qr1”, “Qr2”, and “Qr3” are enclosure (i.e., room) nodes which may contain apertures that connect each other. This OpenFlight model organization is also used in the experimental TerraVista MES import capability.

A custom software tool developed by SAIC takes this OpenFlight model hierarchy and generates an MES correction file which contains enclosure and aperture information. Our tool also makes a number of checks on the data to make sure that the resulting correction file is valid. For example, the program checks that all apertures contain valid enclosure names, that an exterior enclosure exists, that all polygons contain geometry which faces to the enclosure interior (except for the exterior enclosure), and other hierarchy details for a valid MES correction file.

The visual models were then imported into TerraVista to generate a new OpenFlight database with shell MES buildings. The resulting visual can be seen in Figure 4.

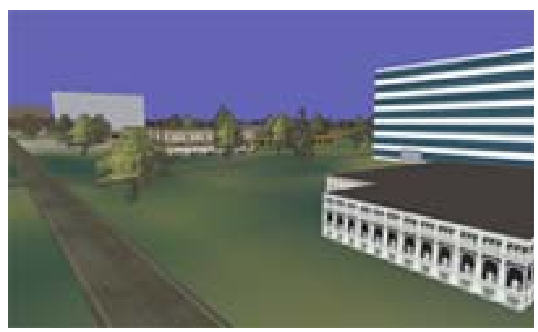

Figure 4 - Visual with simple building models.

The MES corrections file was then incorporated into the CTDB and a database with a small number of MESs was generated, similar to that shown in Figure 5. The OpenFlight and CTDB database with shell MES buildings were used as an intermediate drop for correlation, evaluation, and feedback purposes.

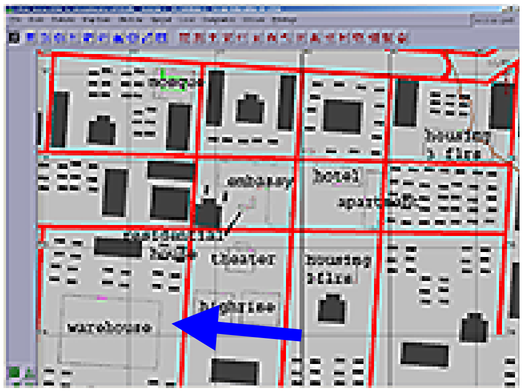

Figure 5 – DISAF CTDB with simple MES.

The next step was to increase the fidelity by creating the interiors for each building that match the JCATS floor plan layouts. We developed a custom program that generates nearly all of the OpenFlight representation of the JCATS building floor plan automatically. This program was the result of two approaches that were attempted to solve the conversion problem.

The first approach to creating an OpenFlight visual model from the JCATS information was to take the simple JCATS 2D representation and generate floor polygons for all the rooms. Note, one JCATS wall line segment (e.g., an exterior wall) may be used to define multiple rooms. The conversion is accomplished by loading a binary space partition (BSP) tree (Graniela 2001b) with the JCATS building floor plan information in a manner that subdivides the lines that define the building rooms. This processing produces a set of enclosure floors lines that define closed regions at the leaf nodes of the BSP tree.

The lines defining the room floor are next converted to a polygon. These room floor polygons are extruded into 3D enclosures and doors and windows apertures cut into the walls. However since the data from JCATS is inherently 2D, additional parameters are needed to generate the windows and doors. For example, window vertical placement and height are added as parameters using reasonably assumed values.

Sorting, clipping and storing the JCATS line data proved to be a much bigger task than anticipated. Although this approach promised to be fully automatic, project constraints made it necessary to adopt a different approach that could be completed more quickly even if it was not fully automatic.

This second approach expands the one wall lines of the JCATS representation to two lines based on a wall thickness parameter. These walls are then converted into vertical planes to form the interior walls. Windows are converted from 2D line to 3D windows and the faces are made coplanar with the walls. An example OpenFlight 3D model for a building model constructed using this algorithm can be seen in Figure 6.

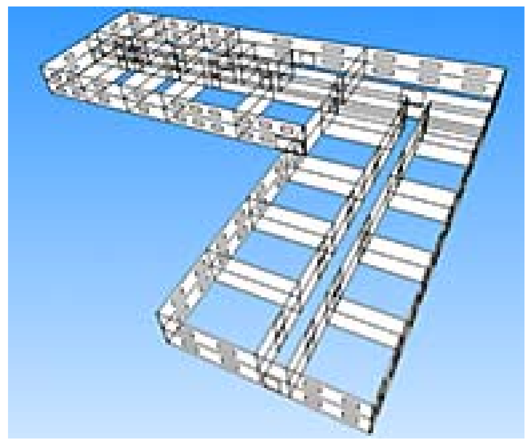

Figure 6 - Extruded walls generated from JCATS.

At this point about 80% of the conversion is done; from here it is necessary to manually edit the models to separate the geometry into enclosures and to cut the windows and doors into the walls. Later, the exterior enclosures are added based on the exterior wall geometry and height of the building. All exterior doors and windows are added to the model. All faces containing surfaces are then cut in and windows and door aperture connectivity information is added to the model.

A large number of MES buildings were desired for the final version of the urban database. We needed an automatic way of generating “OpenFlight To MES” and “recompile” commands for each building. An ArcView script was developed to generate OpenFlight to MES commands as well as “recompile” commands. Each “OpenFlight to MES” command contained the MES coordinates and orientation for the specific building. Since we intend to perform dynamic terrain damage modeling per building, each building must have its own template. In the past (SAIC 2001), multiple similar MES buildings shared the same template. Damage to one building would result in damage to all buildings using the same template. To solve this problem, the "OpenFlight to MES" tool was modified to generate an extra command line argument to generate different template names. The building ID was attached to the model name to make unique template names. Due to the large quantity of MESs that were to be recompiled into the CTDB, and due to limitations with the operating system command line, it was necessary to generate multiple recompile commands. Intermediate CTDBs with an increased number of MES were created at each step until a CTDB with 794 MESs was created.

Among the biggest challenges in the development of this complex MES CTDB environment were: 1) the addition of a large number of MES to the CTDB, and 2) the generation of 3D geometry from a JCATS 2D representation. This project also required modifications to the OTB recompile tool as well as TerraVista CTDB output components. The resulting major achievement was the development of a process which allows the incorporation of a large number of MES into a CTDB.

Mountain Database

The mountain environment is primarily characterized by underground structures similar to that shown in Figure 7.

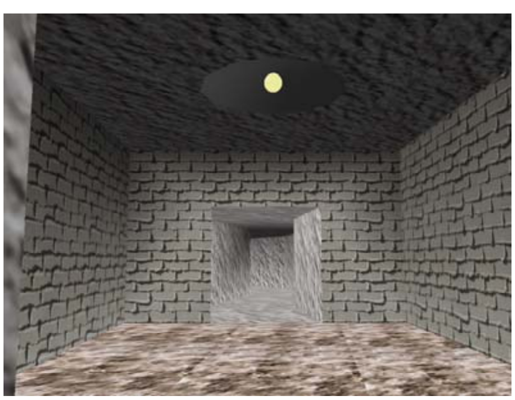

Figure 7 - Mountain tunnel bunker.

The geo-specific part of the world that is being used for this database is a mountain in central Asia. The database contains three underground cave/tunnel complexes that were modeled as MESs. Each cave/tunnel complex contains a set of interconnected hallways with rooms and a number of access holes, portals, or cave openings to the surface, similar to those shown in Figure 8.

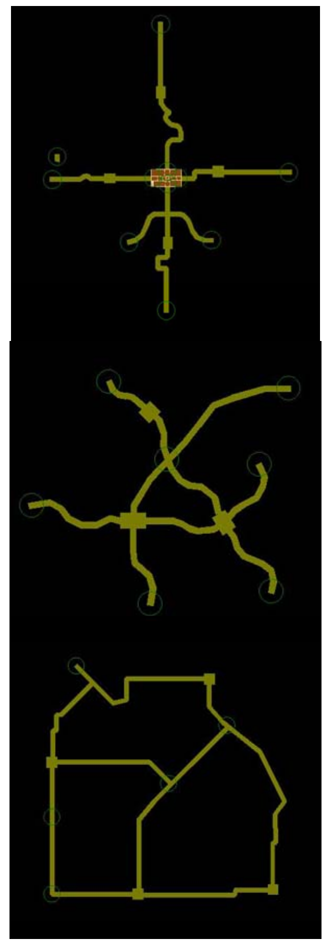

Figure 8 - JCATS tunnel networks.

Leveraging the work done on the development of the urban database, the software which extracted the 3D buildings from the JCATS 2D representation was modified to generate tunnel geometry from JCATS data. Once again the 2D representation structures used in JCATS were augmented with information that could be used to generate 3D geometry. Each tunnel segment was treated as an enclosure with an aperture at each end. Enclosures in

MESs are closed volume regions whose only interface to other enclosures is through apertures. These 3D tunnels were 1.5 meters wide x 2 meters high and were extended based on the shape and dimensions of the caves. At this point the 3D models were textured and attributes added so that they could be converted into MES correction files.

The cave entrances were modeled as separate MES components so that they could be extracted and used as footprints for the tunnel entrances in TerraVista. This allowed TerraVista to build terrain around the cave openings and include this terrain in the visual and CTDB terrain skins. The generated CTDB contained the access portals but not the tunnel structures themselves. At a later time the MES tunnels were then recompiled into the CTDB as correction files. The visual model of the tunnels contained light fixtures which provided light and dark areas in the tunnel. No CTDB representation of this environment feature is possible at this time in the CTDB, so lights were not included into the MES models. However achieving a satisfactory lighting and illumination representation on SVS 3D visual turned out to be a significant challenge. Differences in the runtime implementation of lighting between the 3D modeling tool (Creator) and the SVS required numerous modeling iterations.

Another significant challenge was the shape and inclination of the tunnels entrances. The problem arises because the CTDB recompile tool will not allow holes in the terrain and will add polygons to close them. A continuous terrain skin, will cover the tunnel entrance and block movement and line of sight between the tunnel and points above ground. The workaround we developed was to: 1) create a "well" with nearly vertical sides in the terrain skin underneath the vertical airshafts (see Figure 9), and 2) cover the tunnel openings with a nearly vertical terrain skin polygon (Figure 10). DISAF had previously been modified (Graniela 2001a) to allow line of sight from one point in an MES (tunnel) to another without considering the terrain skin; this permits line of sight from the air shaft to an adjoining tunnel through the vertical polygons forming the sides of the well. DISAF does not normally check for collisions with terrain polygons during movement, so entities can usually move through the vertical terrain polygons covering the entrances.

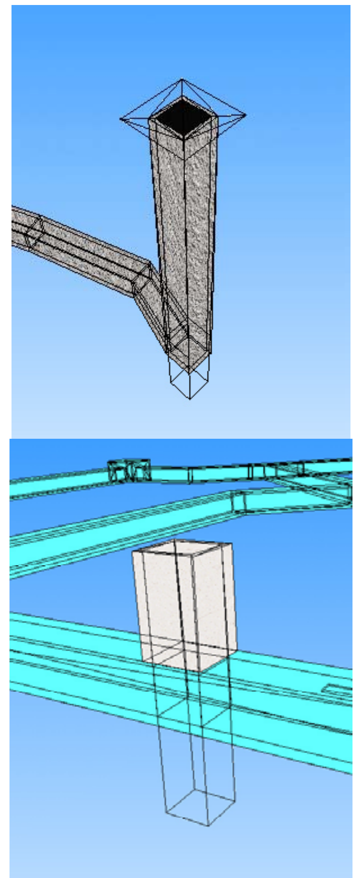

Figure 9 - Cave / tunnel airshafts.

This approach successfully generated a CTDB but access to the tunnel by SAF was not possible in some instances. Mobility from the inside of the cave to the terrain skin worked all the time, but not the other way around. Access to the caves through the airshafts from the outside was always possible, but access through the cave entrances did not work as well. Investigation into a better solution is still on-going. A workaround to this "entrance-passage" issue was achieved by altering the entity's speed and approach location.

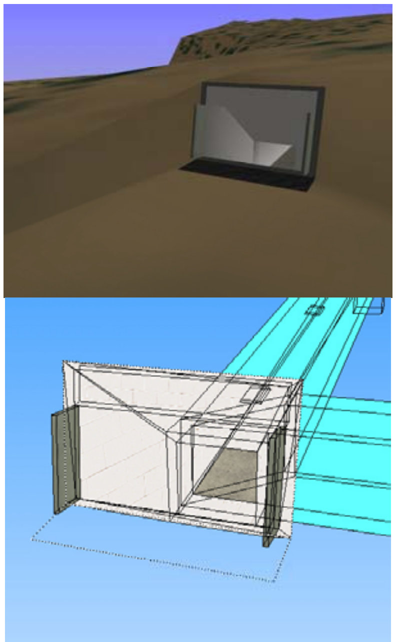

Figure 10 - Cave / tunnel entrances.

Jungle Database

Requirements for a third database, the "jungle" SNE, are currently being developed. Therefore, we will only be able to talk about the conceptual design approach for the development of this database.

Jungles are characterized as rainy, humid areas with heavy layers of tangled, impenetrable vegetation. Tropical rain forests consist mostly of large trees whose branches spread and lock together to form canopies. The rain forest can be described as composed of four layers; 1) emergent layer, 2) canopy, 3) understory, and 4) forest floor.

The tallest trees can be found on the emergent layer. Trees in this layer are typically 60 meter high with trunks that measure up to 5 meters wide. The root systems of these trees are large, as they must sustain high winds. The canopy layer extends beneath the emergent layer as seen in Figure 11.

The canopy, which rises to a height of 45 meters, filters approximately 80% of the light, preventing transmission to the forest floor below. Canopies can exist at two or three different levels, and can be from 20 to 30 feet thick. Air observation is nearly impossible and hanging vines are common. The understory is the area under the canopy which receives 2% to 5% sunlight. The understory is 3.5 meters high. Finally the forest floor receives 0% to 2% sunlight and 100% humidity. The floor is composed of a litter of rapidly decomposing vegetation and extensive above-ground root systems. Ground observation is limited to about 50 meters. Vehicular travel is difficult, whereas foot movement is easier. This is the description of the typical jungle which we are developing. Of course limitations on the representation and rendering of this kind of environment will limit its modeling.

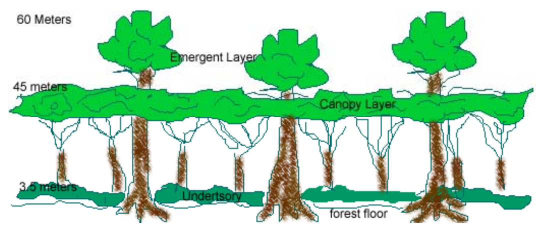

Figure 11 - Jungle canopy.

The jungle database is based on a geo-specific location somewhere in south west Asia. The database environment will contain two underground tunnels, one village, and have dense jungle-like vegetation. Once again JCATS data will be available. It will be used to model the village, tunnels, roads and terrain skin.

The database design will represent a jungle environment with a complex array of trees, trees lines and tree canopies similar to that shown in Figure 12. Large trees which emerge from the canopy will be randomly distributed though out the forest. Large tree trunks will be used for these trees. Each tree will be shaped so that SAF and SVS soldiers can move close to the tree trunks. "X-shaped" tree visual models with variable trunk and foliage widths will be used.

Figure 12 - Jungle vegetation prototype model.

Double sided horizontal polygons will make up the forest tree canopy. This canopy will cover the entire forest and will limit visibility in both directions, allowing only partial visibility from the air and from the ground. In CTDB the "tree fullness" attribute will be used to vary the foliage density representation. The resulting databases should have various degrees of horizontal and vertical visibility throughout the jungle. Tree lines will be used to model the understory layer. They will add to the complexity of the environment and will slow mobility. Figure 12 shows an image of a prototype model that was developed as a proof of concept. The model produces a realistic 3D environment of a jungle environment.

SUMMARY

This paper has presented some of the techniques currently being used at SAIC for the development of visual and SAF SNEs. These SNEs provide adverse environments for evaluation of FCS systems, especially those involving dismounted infantry. Our development of these SNE leverages past experience.

The urban, mountain and jungle SAF SNE contain complex MES structures which allow dismounted infantry to enter buildings and caves and operate realistically in battlefields that are dense with natural and man-made features. The generation of correlated visual and SAF SNEs for dismounted infantry requires simultaneous development of models for structures with interiors. Limitations on the SAF SNE representation limit the amount of detail to be included in the visual model features. For the generation of correlated SNEs, the geometry and features for both visual and SAF databases must agree to guarantee a fair fight.

SAIC will continue developing such environment pairs while improving the development process to achieve cost effective and rapid development of correlated SNE.

REFERENCES

Boeing WWW, http://www.boeing.com/defense-space/ic/fcs/bia/flash.html

Graniela, B., Pigora, M., and Reece, D. (2001a) "Urban Human Simulation Environments in CTDB". In *Proceedings of the 2001 Spring Simulation Interoperability Workshop*. IEEE.

Graniela, B., Cisneros, J., and Reece, D. (2001b) "Synthetic Urban Environments Using Quake II Engine". *Interservice/Industry Training Systems Conference.* American Defense Preparedness Association.

Lawrence Livermore National Laboratory, 2002, JCATS 4.0 Terrain Format.

MultiGen WWW, http://www.openflight.org/support/dc_standards.shtml

SAIC, DISAF Capabilities Enhancement Year 3 (DCE3) Final Report, 13 Dec 2001, SAIC Document Control # SAIC-01/7569&00, Contract # N61339-01-C-0045.

SAIC WWW, http://www.asset.com/solutions/simsystm/disaf.html