Improving the Fidelity of Transportation Features in Simulation

Kevin Bentley
Cognitics, Inc.
Boise, Idaho
kbentley@cognitics.net

Jim Pivonka
SAIC
Orlando, Florida
james.w.pivonka@saic.com

Mark Johnson
PEOSTRI - SE Core
Orlando, Florida
mark.doyle.johnson@us.army.mil

ABSTRACT

As the overall fidelity and performance of simulation systems increases, there is a need for improved representation of transportation features. Roads, bridges, tunnels, and other drivable features need to be accurate representations of the real world and interoperable across a confederate of training systems. Improvements in the fidelity of these features in the past have primarily been focused on increased texture resolution and integration of hand modeled features. However, as the capabilities and expectations of ground based training systems increases, there is a need for major upgrades to the representation of transportation features.

This paper details the research and implementation efforts to greatly improve the appearance, and functionality of transportation features on simulation systems. The goal of this effort is to model detailed geospecific roads, bridges, and tunnels that can be used in virtual and constructive simulation systems.

Road networks and associated features have typically been represented as two-dimensional GIS linear features with attribution. However, in reality, roads are much more complex. Roads can vary in width, number of lanes, and driving rules at any point along the road. Placement of traffic lights and signs requires knowledge of the traffic laws and lane types for each lane in a road. Complex intersections such as freeway interchanges often involve complex three-dimensional interactions between roads where some roads merge, some roads intersect, and some roads pass over or under other roads. Because of this enhanced data models were required to represent these complex features.

Our solution includes a standalone topology analysis tool that converts the GIS data features to an optimized OpenDRIVE format that is capable of representing road geometry and topology. A model generation toolkit can read this format and output textured geometry for a wide variety of simulation tools and terrain generation platforms. Finally, we describe the integration with a production database generation program's workflow.

ABOUT THE AUTHORS

Kevin Bentley is the founder and president of Cognitics, Inc. He has over 20 years of experience developing software for GIS, simulation, and video games. He currently is the principal investigator for a Phase II SBIR focused on research and development of technologies that improve the representation of transportation models that integrate with terrain in simulation systems. In addition to commercial video game development experience, Mr. Bentley has extensive geospatial and GIS experience, covering areas such as high precision GPS, commercial and open source GIS tools, and APIs.

James Pivonka is a senior software engineer at Science Applications International Corporation in Orlando, Fla. He is currently supporting the Synthetic Environment (SE) Core Common Virtual Environment (CVE) program with terrain database generation system software development and architecture. Mr. Pivonka has over 13 years of experience in the simulation field having supported large programs such as CCTT, UKCATT, OneSAF as well as working for commercial off the shelf (COTS) terrain database generation system companies.

Mark D. Johnson is a senior systems engineer at Strong Point Research in Orlando, Fla. He is currently supporting PEOSTRI as a SETA contractor, a technical expert in visual systems, synthetic environments, software development and architecture. Mr. Johnson received his undergraduate degree in computer science from the Utah State University. He has been working with simulation and training systems since 1987 and has more than 35 years of experience in software and system design.

BACKGROUND

Procedural modeling of transportation features is desirable in simulation for the ability to model large areas of the world with minimal human input. Synthetic environments can be created quickly compared to hand modeling each bridge, overpass, or highway interchange. With procedural modeling, multiple correlated representations can be made automatically, each with a different level of detail based on the targeted runtime. As capabilities of a runtime improve, the same area can be regenerated at a low cost, simply by changing the procedural generation parameters.

In order to effectively model high fidelity transportation models in synthetic environments, several technical obstacles must be overcome. This paper describes a multi-year effort that included small business research under the SBIR program (Brinton & Bentley, 2011), as well as integrated technical teams that involved both large prime contractors and the government. This effort was centered on solving some of the many complex problems that have held back the fidelity of transportation features in simulation. The following section describes some of the major challenges we encountered, followed by the approach we took to mitigate or resolve those challenges.

This effort has been focused on improving the representation of transportation features in simulation databases. A major goal was to address anomalies found in current tools and to increase the fidelity of transportation features in order to improve the training experience. This paper describes the generalized process and solution with special attention towards the end of the paper to integration with the CVE program.

MAJOR TECHNICAL OBSTACLES

Obstacles in Data

A major obstacle when modeling transportation features is the lack of information. Geospatial data products are readily available, but on closer inspection a great deal of information about features such as roads is missing. Much of this data is unnecessary for generating simplistic models of roads, but becomes critically important when modeling roads with higher fidelity.

Most transportation data products convey only two dimensions of the roads shape. However, in the real world roads vary along their length. Attributes such as the width of the road, the number of lanes, or the relationship with the ground may change either suddenly or gradually along the length. Each road may have a different configuration of lanes, lane widths, shoulders, surface type, etc. GIS data sometimes has attributes that specify the number of lanes for a road, or the width of a road, but there is not a simple way to convey these values that change within a road. A GIS feature may represent two roads as crossing, but what is the relationship between these roads? Do the roads intersect? If so, what type of intersection is it? Does one road go under the other, or over it? Most if not all of the answers to these questions are unavailable from common GIS road data.

Even when GIS data has vertical information about a road, this information is commonly represented as an integer value of how many levels above ground the road is raised. This information does not convey where on the road, or where on a connected road the slope changes to transition from one level to the next.

Obstacles in Fidelity

Another obstacle comes from mismatched fidelity in data. This mismatch is usually most dramatic when comparing the digital elevation model (DEM) that represents the shape of the ground with the vector data that represents the position of the road. Digital Terrain Elevation Data (DTED) level 1 is commonly used to build terrain databases. This data has a horizontal spacing of about 90 meters. A typical road that is about 10 meters wide will not be accurately represented in the DEM data. This difference in fidelity causes roads to have unrealistic slopes and banks that must be corrected through cuts and fills when building terrain. Applying these cuts and fills requires information about the topology (i.e. the relationship between roads) in order to model intersections with other roads correctly. Without the topology, inappropriate cuts and fills can make the terrain worse. For example, a fill might be made to bring the slope up to civil engineering standards, but this may cause a more severe error in an adjacent road. Only by analyzing nearby and connected roads can cut and fill decisions be properly made.

Obstacles in Runtime

Our research and development effort has not been focused on solving problems for just one specific simulation program or system. Instead we have focused on creating new data models and software capabilities that can be applied to both constructive and virtual simulation. This means supporting many different simulation programs involving integration with systems produced by different vendors, each with their own capabilities and limitations.

Runtime systems generally are concerned with either the visible representation of transportation features (e.g. textured triangles or other surfaces), the logical representation (e.g. a network of roads), or often both. To be useful, these representations must correlate and not conflict. Correlation is a difficult problem to solve when two systems represent a feature at a different fidelity, but it can be improved if the two representations are generated using the same algorithms and the same original data.

Different runtime systems using the same source data sometimes have conflicts and limitations because of the different data models. One runtime system may represent roads as attributed polygons with a special surface material code, while another system may represent roads as a simple two-dimensional string of points. A complete solution must be able to create multiple data products for different systems while maintaining the correlation and topology between these data models.

Within the virtual simulation domain there are a great deal of differences in data models and capabilities. While some systems are limited in the number of polygons or texture memory that can be used to render the visual representation, other systems are capable of rendering comparably massive quantities of geometry and textures. Because both systems are in use (often within the same program), it is desirable to have a system that can support each system to its maximum utilization. The complexity of a road can be represented by a simple set of textures and complex geometry (where each different lane, marking, and material use their own texture), or it can be represented with simple geometry and a texture that is specific to the exact configuration in a road. In the real world the latter is difficult to achieve because there is not a discrete number of road configurations used in most parts of the world.

OUR APPROACH

Topology

Our initial effort was focused on solving the problem of inferring the 3D aspects of complex highway interchanges from 2D GIS features. It quickly became apparent that we would need a new data model beyond what was available in GIS formats. In order to determine when two roads intersect or pass each other, we would need to construct a network of relationships between roads. We initially created a data model that we could use to store the inferred topology for use in model generation. We refer to topology here as the data that models relationships between logical features. These relationships may be symmetric or asymmetric. For example, two roads may have a relationship of predecessor/successor. One road may have a relationship of ‘passes over’ another road. Or two roads may have a symmetric relationship of ‘nearby’ or ‘parallel’.

The need for topology in model generation is not always obvious. A road network is obviously needed for applications such as path finding, where a computer controlled entity needs to traverse the road network in order to plan a route to a new destination. As we progressed in our model generation capability, the need for topology became clearer, and it became obvious that topological information beyond intersections was required.

When building intersections where roads meet at oblique angles, the topology of all of the roads that participate in the intersection is useful to construct simplified geometry. This same information allows the intersection to be built with sidewalks and bike lanes through the turns of roads. If lane topology information is available it is possible to apply decals on the road indicating which direction traffic can follow. For example dedicated right or left turn lanes can be represented accurately. The same lane topology information can be used to place operable traffic control devices that can be used to control computer generated entities in a constructive simulation.

When two roads are connected but have a different number of lanes, predecessor and successor information allows the automatic creation of merge lanes that can be tapered to avoid abrupt changes in road widths. This is a common visual anomaly seen in many modern simulation systems.

Because of the difference in fidelity of terrain and road representation, cut and fill operations are frequently required to make roads in a simulation drivable and realistic. Creating the cut and fill instructions requires knowledge of how roads interact to avoid anomalies in the road surface. Simple predecessor and successor relationships between roads makes it possible to gradually modify the slope of several consecutive roads without any abrupt changes in the slope. Modifying the elevation of a road may have an adverse effect of nearby (but not directly connected) roads, so information on roads with a 'nearby' relationship allows a more sophisticated and more accurate cut and fill operation.

Figure 1 - Typical Complex Highway Interchange

When inferring the configuration of a complex highway interchange such as shown in Figure 1, having the topology available makes it much easier to determine which roads would be in conflict if all roads were placed directly on the terrain. One of the first steps of our solution's process is to identify all roads that are in this type of conflict so a solution that raises some of the roads to clear the other roads can be calculated.

We are mostly concerned with topology of roads and its constituent parts, but information about relationships between roads and other objects is also useful. The basic topological relationships between roads and lanes are illustrated in Figure 2. Additionally, traffic control devices such as stop/yield signs, and traffic signals have important relationships with roads. When generating models of roads and intersections the lane topology can be used to automatically place realistic and operable traffic lights in intersections. The same data can be used at runtime to control the flow of traffic through those intersections while controlling the visible representation of the traffic signal. This has important application when mixing computer generated entities with virtual training.

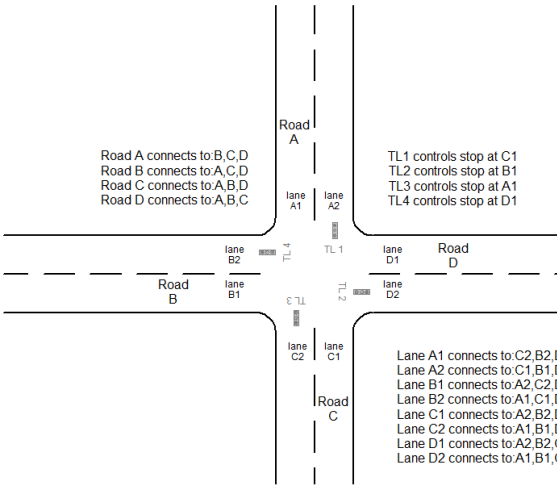

Figure 2 - Basic Topology of Roads and Lanes

Informational and controlling traffic signs can be placed next to roads using this data as well. Other objects including street lights have useful relationships with roads. If lights or signs are placed along the length of a road, this information can be used to automatically adjust the position of the lights if the road is moved.

Other features such as buildings can also have relationships to roads. For example a building can have a relationship to the nearest cross roads to the building. This can be helpful when computer generated entities are path finding, but it also can be useful to communicate directional information to humans.

Enhanced Data Model

One of the primary limitations in GIS data formats is the lack of multi-dimensional attribution. A single feature can have attributes that describe various characteristics of the entire feature, but it cannot describe attributes for different parts or dimensions of a feature. For example, a road may have two lanes for the first 200 meters of the road and then split into four lanes later on.

Our initial data model allowed each vertex in a line that defines a road to have a full set of attribution. With this approach, we could define the widths, the number of lanes, surface characteristics and more through attribution for each vertex that defines the geometry of the road. However, this approach became cumbersome when we applied operations to the roads that added or deleted vertices.

To define the relationship data model, we identified several types of relationships, and we assigned a universally unique identifier (UUID) to each feature. Each road would then store a list of UUIDs for each type of relationship. For example, a road may start at a start or end point of one or more other roads. In that case, the road would have a list of UUIDs for the 'start at' roads.

As we progressed in the identification and definition of the characteristics and variations that exist in real-world roads our data model became more and more complex. We initially created a proprietary data model and used it to prototype a processing pipeline and model generator that could analyze roads in GIS formats and build textured 3D models for use in virtual simulation. During the 2011 I/ITSEC conference we demonstrated this prototype to several interested parties who inquired about supporting the OpenDRIVE format. After analyzing the OpenDRIVE format in depth we realized that this format addressed most of our core needs, and was extensible enough to allow us to add capabilities as needed.

OpenDRIVE was created in Europe by a team of driving simulation experts (Dupuis, Strobl, & Grezlikowski 2010). It is an open standard with the goal of a logical description of roads. The OpenDRIVE document is an XML file, so it is easily extensible. OpenDRIVE describes roads as being built with lane sections, with each lane section has a fixed number of lanes. Each lane has a width defined by a polynomial function of the length (this allows curved widths that are common in the real world). Lanes have marks that are used to describe the visual appearance of the road marking as well as the driving rules for lane changes. In OpenDRIVE, the geometry is defined as a 2D plan either with vertices (similar to how GIS features are stored), or via spirals or curves. Road elevation data is stored as a profile along the length (so a road can have a varying height independent of the vertices that define the 2D geometry of the road).

OpenDRIVE uses *s* coordinates and offsets to apply properties to different parts of the road. The *s* coordinate defines some position between *0* and the 2D *length* of the road. This allows properties to vary at any point along the road without being associated with a particular vertex. If vertices are added or removed the *s* values will continue to be valid as long as the 2D length of the road remains static. This allows the road to vary independently of the vertices that define the geometry.

We adopted OpenDRIVE to use both as an internal data model while processing the GIS data into logical and geometric output formats, as well as an interchange format. Because OpenDRIVE is an XML format and simulation datasets can be very large, it is not well suited for runtime processing operations due to the large file size and the inefficiencies in parsing XML to find relationships. For the internal data model we created a binary storage format based on the SQLite database engine. We call the binary data an OpenDRIVE Database (ODDB). The ODDB allows high performance indexing which is very important when developing the data processing tools. OpenDRIVE data operates in a Cartesian coordinate system, so we added spatial reference frame tags to the OpenDRIVE schema to support geo-referencing. Finally, we extended the OpenDRIVE data model to represent logical road states through *s*-based attribute tags. This allows the OpenDRIVE data to encode logical relationships such as 'passes over.'

Cognitics, Inc. developed this set of OpenDRIVE functionality for use in simulation under the product name SPAWN. In order to provide greater access to this data and to ease integration, Cognitics has created an open source royalty free C++ library that can be used for reading and writing OpenDRIVE data; both in the native XML form, and in the optimized binary SQLite form.

Stand Alone Topology Tool (STAT)

The process of going from GIS road data to OpenDRIVE data and finally models for virtual and constructive simulation is an iterative, non-linear process. This process requires analysis of many roads simultaneously in order to determine the topology and prepare the data for model generation. We created a software application we call the STAT. STAT is responsible for ingesting GIS feature data, and outputs OpenDRIVE data. The OpenDRIVE data can be used for path-finding of computer generated entities and also for model generation. Multiple representations can be built with the OpenDRIVE data in order to support systems with different formats or different levels of fidelity and complexity. All of these representations will correlated because they will have been generated using the same detailed OpenDRIVE data.

The STAT is built on a set of software modules originally developed by Cognitics under a Phase I and Phase II SBIR project. STAT evolved during integration with the CVE program to meet the particular needs and workflow of the program. The core set of software functionality may be offered commercially by Cognitics in the future. STAT has four major functions: Topology assignment, modeling roads from civil engineering rules, assigning elevation to roads, and generation of cut and fill volumes for terrain near the roads.

Topology Extraction

Determining topology from GIS data may seem like a simple process, simply connecting roads that end or start near another road. In the real world though, these roads may not be connected at all. A ramp that connects two freeways may cross over a local road, even ending directly over the local road. But these roads do not share an intersection. In order to make these distinctions, the STAT uses attribution along with clues in geometry, and civil engineering rules. For every road processed, the ODDB data is searched for nearby roads, relying on spatial indexes to speed the process. When a nearby road is detected, the attribution is analyzed to determine the direction and type of road. For instance, a one-way ramp that touches a freeway in the wrong direction would not result in an intersection. But two local roads that meet at a point would. If two roads meet at an acute angle, this can be an indication that they do not connect. Other rules look at categories of roads. For instance, a freeway should normally never intersect with a local road (where a local road appears to end near a freeway, it is likely an over or underpass), but it may connect to a local road via a road flagged as a ramp. At the end of the road linking process (after all roads have been processed) individual lanes are connected within and between roads to allow the OpenDRIVE data to specify driving rules and traffic control devices at a very fine level of detail).

Modeling Roads

Even when detailed attribution such as that found in Navteq data exists for roads, there are many variables left to determine. In the United States, several publications such as the *Manual on Uniform Traffic Control Devices: For Streets and Highways* (United States, 2009) and *A Policy on Geometric Design of Highways and Streets: 2004* (AASHTO, 2004) exist that define standard road parameters for construction. These parameters included things like the width of lanes and road marks as well as the length of merge lanes. While there are standards, these parameters can still vary between different types of roads, and different regions, even within a state or country. We created a data model to store these parameters in a Civil Engineering Ruleset (CER) document that allows rules and parameters to be both regionally specific and vary based on types. When augmenting the GIS road data to produce OpenDRIVE data, the STAT reads the CER file and queries it for each road being built. This way, each road is built using civil engineering rules that are specific to the road type and location.

Elevation Assignment and Conflict Correction

An important process in the STAT is the assignment of elevation (Z) values for each road and intersection. The STAT reads a collection of digital elevation model (DEM) files and queries it for each road to build a road profile. Initially each road is draped on the DEM. Once the draping is complete, each road is checked for conflicts. Conflicts are areas where the 2D outline of a road intersects with the 2D outline of another road. In a normal intersection between two roads the outlines will not intersect because the topology indicates that the roads are connected. In this case they are modeled to start and end in a seamless connection. In the case where a local road crosses a freeway a conflict exists because both roads initially are draped to the same elevation value. This process identifies groups of roads that have conflicts to be resolved.

In a simple case of a local road crossing an interstate, the elevation optimizer process in the STAT can simply raise the elevation of the local road and mark it as having ramps and a bridge over the freeway. When more complex intersections such as those seen in Figure 1 exist, the solution is non-trivial. During our initial attempts to solve these problems using intersections in Oahu Hawaii, we found there were too many permutations to calculate all of them to find an ideal solution. For example, if there are only two distinct elevation levels (one on the ground and one raised) to consider and 40 roads involved in a conflict, there are 2^{40} or approximately *one trillion possible combinations* to consider. In several highway interchanges we have modeled there are three or four distinct elevation levels in the real world. In order to tackle this problem we developed a genetic algorithm to find an optimal (or nearly optimal) solution quickly. Our cost function considers the cost of modifying a road (by raising the elevation above ground) based on the width and length of the road as well as the impact such a change has on the road's neighbors. The genetic algorithm allows us to converge on a near optimal solution much more quickly than a brute force analysis.

Cut and Fill

The final step in the STAT process is the fine tuning of road bank and slope, and the creation of terrain cut and fill volumes. This process uses the road outline built with the OpenDRIVE data and overlays it on the terrain. A new GIS vector layer is created with an outline for each road with elevation values set to establish the bank and slope of the roads. In order to perform this process so that roads are smoothly connected, the topology information in the OpenDRIVE data is used. This ensures that road and terrain slope changes gradually whenever possible, using civil engineering rules found in the CER. After the road base polygons are built, the cut and fill process analyzes the side slope of the terrain surrounding roads and creates cut and fill polygons that are also added to the output GIS layer. After this process completes, the output GIS layer is fed into the terrain database generation system (TDBGS) and the vector features are used as constraints when building terrain polygons. The basic cut and fill operations are illustrated in Figure 3, where cuts (removal of terrain) and fills (addition to the terrain) can be seen.

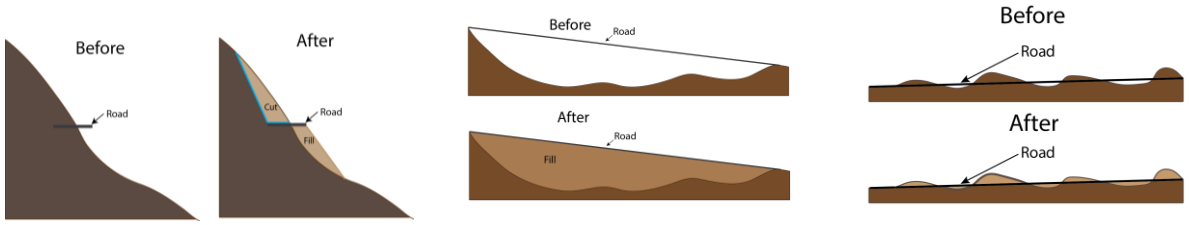

Figure 3 - Cut and Fill Operations (Left: Side Slope, Middle: Forward Slope, Right: Undulation Removal)

INTEGRATION

Tool Integration

The integration into an existing terrain database generation system (TDBGS) was accomplished both by configuring it to process the new data with existing functionality and writing new modules to replace existing ones. The simplest task in tool integration was configuring the TDBGS to interpret and generate the cut and fill geometry identified by the STAT with its existing functionality by a configuration change. Integration of the functionality to generate high fidelity transportation geometry models from OpenDRIVE data was accomplished by the creation of two software modules. One module generates the road and junction geometry whereas the second generates the tunnel and bridge models. The two geometry generation modules have their own challenges.

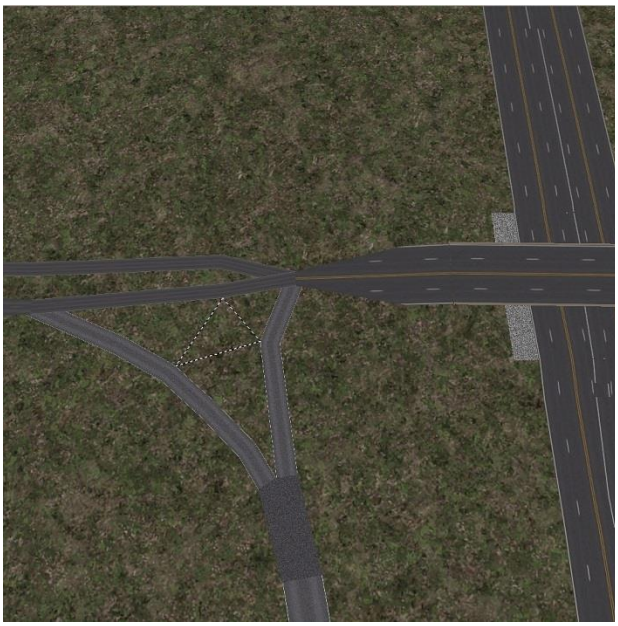

Figure 4 - Geometry from Original Road Module

One of the major challenges of the road and junction module is that OpenDRIVE allows the modeling of very complex roads. Roads can be made up of lanes of varying sizes, have one or more turn lanes, merge and taper lanes and shoulders. Junctions where roads meet can be equally complex. The original module for creating roads simply expanded the linear feature based on the width attribute and applied a road texture to the resulting polygons. For junctions it simply approximated a polygon that allowed the roads to join not taking into account how the lanes that made up the joining roads join each other. Figure 3 illustrates the typical geometry that resulted from the original road generator. In order to take advantage of the OpenDRIVE data and increase the fidelity, two solutions to handle the generation of complex geometry were undertaken.

The initial solution procedurally created polygons for each of the road components. This meant separate polygons were created for the lanes, lane and traffic markings, shoulders, sidewalks, curbs and merge and taper lanes. Figure 4 illustrates the geometry that resulted from this approach. The polygon count with this solution can be high but the tradeoff is an incredibly high fidelity road network. This solution will be able to support driver trainers such as the Common Driver Trainer which requires such detail to roads but the high polygon count can adversely affect the runtime systems used by programs such as the Close Combat Tactical Trainer (CCTT). To support the existing programs in addition to future ones a solution that also supports a lower polygon count was required.

In order to generate roads with polygon counts closer to the original road module but preserving the overall size and shape of the high fidelity approach a hybrid solution was devised. Instead of procedurally creating everything the linear is expanded like the original road module and a texture that closely matches the OpenDRIVE data is applied. For example if the road is a two way road with four lanes that texture is applied. In the case of turn, taper and merge lanes that geometry is created to augment the road and a default material is applied based on the road's surface material such as concrete or asphalt. The junctions are formed such that they take into account the connectedness of the lanes that make up the roads but they do not have any markings and use the same default texture as the turn, merge and taper lanes do. This approach was able to reduce the geometry but yet maintain the more complex shapes of the first approach resulting in an overall more realistic road network where the interactions of lanes are taken into account. A third solution to accurately generate high fidelity roads and still minimize the geometric complexity is being explored. The third solution is to procedurally create textures for the roads thus shifting the complexity from polygon to texture space. As of this writing, the third solution is still being explored.

Figure 5 - Geometry from New Road Module

The second module created to support the higher fidelity transportation networks was the bridge and tunnel module. To support the procedural creation of complex bridges and tunnels that can allow for regional differences the concept of model kits was introduced. The model kit is a collection of cross section models that are expanded along the length of the linear feature that defines the bridge or tunnel by the bridge and tunnel module. The Standalone Topological Analysis Tool (STAT) ensures that the road networks join with the bridge decks and tunnel floors through its elevation optimizer. For bridges it must raise up the roads to form ramps. For tunnels the task of STAT is a little more complicated in that it must also ensure that the tunnel is below the terrain except at the entrances. For bridges the bridge and tunnel model just has to instance the new bridge model it created and thanks to STAT it will form a watertight connection with the road geometry. For tunnels there is a complication in that it must cut holes in the Triangulated Irregular Network (TIN) that makes up the terrain and ensure that the terrain forms a watertight seal around the tunnel. Both bridges and tunnels are represented by external models which allows for the support of state models such as damageable ones.

CVE Workflow Integration

The CVE program builds simulation data for a number of simulation programs using a suite of software tools and products. The goal of this integration was to improve the capabilities of the program, and to allow for increased fidelity in transportation features. Many of the initial requirements were derived from defect reports due to anomalies in the representation of transportation features. Many of these anomalies can impact training significantly, and are not easily resolved with the then current set of software tools.

The integration of the new approach for improved fidelity into the CVE workflow impacted three areas: GIS analysis, model creation and database generation. It begins with GIS Analysis. To support the generation of higher fidelity transportation features additional data needed to be collected during feature extraction as well as how features are extracted. The additional attribution included such information as the number of turn lanes, the relative height a bridge or tunnel to its nearby transportation features, whether it has sidewalks and so on. The CVE program currently has a Runtime Database Generation Toolkit (RDGT) that consists of A Commercial off the Shelf (COTS) software as well as software written specifically for the needs of the program. Because the RDGT is very modular, we were able to target specific pieces of functionality for enhancement during this effort. Previously the database generation system, a COTS application still in use by CVE for many other processes, determined where bridges were but in the new extraction guidelines the GIS Analysts are the ones who identify which features are bridges which results in more accurate source data.

Previously the source data that was processed by the GIS Analysts would be handed directly over to the Database Engineers who would have the terrain database generation system ingest it and immediately begin building databases. In the improved process, the source data is processed through the Standalone Topological Analysis Tool (STAT) which performs a detailed analysis and augments the source data so that a high fidelity transportation network can be created with the new geometry generation modules. An unexpected result of running the source data through STAT is that certain errors in source data extraction were found such as two one way roads dead ending into each other. A report capability was added to STAT to perform a quality assurance check where it checks to see if the GIS data makes logical sense and then generates reports that can be used to both correct the source data and improve the GIS extraction process by providing quick feedback regarding the quality.

To support higher fidelity bridges and tunnels, model kits are now required. To support this, a Model Kit Editor was created and the modelers were given training on how to use it to create model kits. The modelers start with a geotypical model of the bridge or tunnel which was modeled using their normal modeling software. The model is then imported into the Model Kit Editor where a series of cross section models are extracted from it. Then the cross section models are assembled into a model kit. The database engineers take the model kits and use them in their configuration of the terrain database generation system. This allows for the generation of complex bridges and tunnels as well as being able to regionalize the models to specific areas within the databases.

The terrain database generation system used by CVE supports a modular approach. As such, the existing road and bridge modules were disabled and the new Road and Junction and Bridge and Tunnel modules were incorporated. To support higher fidelity transportation networks the new modules also require more parameters. This resulted in a more complex configuration of the terrain database generation system on the part of the database engineers.

FUTURE WORK

Future efforts are planned to integrate rail features and hydrology features with roads, and to continue to improve the tradeoff of complexity in geometry vs. increased texture memory. Procedural creation of damaged and destroyed bridges, tunnels, and roads are also planned, as well as integration of detailed traffic control devices. Dramatic improvements to the fidelity and realism of transportation models are possible, not only through database generation but also through behavioral models that operate on transportation networks. We have collaborated with efforts addressing run-time capabilities, such as that described in (Watkins, Moerk, Tamash, Overman, and de la Cruz, 2013). This effort is closely related to and complementary with the research described in this paper, and provides an overview of future improvements that can leverage improved road network data to build more complex computer generated forces models that follow civilian traffic rules.

ACKNOWLEDGEMENTS

The authors would like to thank NAVAIR for supporting this effort through related Phase I and Phase II SBIR projects, and the US Army SE Core CVE program for their support in adopting this technology into their process. Additional support through US Army RDECOM STTC has enabled integration of this technology into the RUGUD database generation system, which enables this technology to reach a wider range of simulation systems. Finally we would like to thank Diamond Visionics for their support in integration of the SPAWN model generation system to the Genesis RT image generation system.

REFERENCES

American Association of State Highway and Transportation Officials., & American Association of State Highway and Transportation Officials.(AASHTO) (2004). *A policy on geometric design of highways and streets: 2004*. Washington, D.C: American Association of State Highway and Transportation Officials.

Brinton, A., & Bentley, K. (2011, September 12). *Modeling the real world - procedural model generation for training and mission rehearsal*. Retrieved from http://www.cognitics.net/modelingtherealworld

Dupuis, M., Strobl, M., & Grezlikowski, H. (2010). OpenDRIVE 2010 and Beyond – Status and Future of the de facto Standard for the Description of Road Networks. *Proceedings of the Driving Simulation Conference DSC Europe 2010*, 231-242.

United States. (2009). *Manual on uniform traffic control devices: For streets and highways*. Washington, D.C.: U.S. Dept. of Transportation, Federal Highway Administration.

Watkins, J., Moerk, J., Tamash, T., Overman, R., & de la Cruz, J. (2013). *Simulation of Civilian Traffic Rules in Computer Generated Forces*., I/ITSEC 2013, Orlando, FL.