

## Automated Modelization in Terrain Database Production

**Stephen Eckman**  
GameSim  
Orlando, FL  
stephen.eckman@gamesim.com

**Ronald Moore**  
Leidos  
Orlando, FL  
ronald.g.moore@leidos.com

**Mark Johnson**  
U.S Army PEOSTRI  
Orlando, FL  
mark.d.johnson302.ctr@mail.mil

**Jaeson Munro**  
PAI  
Orlando, FL  
jaeson.h.munro@leidos.com

### ABSTRACT

The feature content requirements for terrain databases used in the Modeling and Simulation industry continue to grow. The requested database geographic extents are expanding. Database feature densities and complexities are increasing. New simulation systems are leveraging game engines, enabling this growth in database content requirements. Unfortunately, the cost of producing 3D cultural models, that reflect these increased complexities, using traditional commercial 3D modeling tools, by hand, is quickly becoming unsustainable, and maintaining and updating 3D cultural models libraries are becoming unaffordable. The process of assigning 3D model references to vector features in a geospatial database, identified as modelization, has been automated by the SE Core program to significantly reduce database production costs, improve database quality and consistency and increase run-time database performance. This paper describes the processes and tools used in the Automated Modelization Process in the production of the Synthetic Environment Core (SE Core) terrain databases. When applied, these tools and processes results in a cost effective approach to automatically generate 3D cultural models that are designed for the specified run-time requirements, saving considerable development time when compared to using commercial 3D modeling tools and manual processes.

### ABOUT THE AUTHORS

**Stephen Eckman** is the Chief Scientist at GameSim. Mr. Eckman has 14 years combined experience in the modeling, simulation, and gaming industries. He joined GameSim in April of 2012 and has led multiple research efforts for NASA, the Air Force Research Laboratory (AFRL), and the Army Research Laboratory (ARL). He is currently leading the development of the Procedural Model Generation Service for ARL. Previously Mr. Eckman was the Chief Engineer for the On-Line Interactive Virtual Environment (OLIVE) at SAIC. Before working on OLIVE, Mr. Eckman led numerous virtual training projects and applied his specialization in telecommunications to the video game industry, where he was the lead developer for a massively multiplayer space exploration game, and added online features to the Madden series of video games. Mr. Eckman has a bachelor's degree in Computer Science and also holds a certificate in Gamification.

**Ronald Moore** is currently the Chief Architect on SE Core CVEM. He has over 30 years of experience in the simulation and training industry with expertise in software development, computer graphics, computer image generation, simulation geospatial database production, video, audio, sound simulation, and PC and console game development. His previous assignments include Senior Systems Engineer at SAIC; Research Scientist at E&S; Chief Technology Officer at Infogrames, and Lead Software Engineer at Boeing. Ron holds a BSE from BYU.

**Mark Johnson** – Mr. Johnson is a senior systems engineer at Strong Point Research in Orlando, FL. He is currently supporting PEO STRI 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 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.

**Jaeson Munro** is currently a software developer for the SE Core CVEM, He has over 15 years of experience in the simulation and training industry, primarily focusing on developing tools for terrain database production. Mr. Munro received his undergraduate degree in computer science from The University of Central Florida.

## **Automated Modelization in Terrain Database Production**

**Stephen Eckman**  
**GameSim**  
**Orlando, FL**  
**stephen.eckman@gamesim.com**

**Ronald Moore**  
**Leidos**  
**Orlando, FL**  
**ronald.g.moore@leidos.com**

**Mark Johnson**  
**U.S Army PEOSTRI**  
**Orlando, FL**  
**mark.d.johnson302.ctr@mail.mil**

**Jaeson Munro**  
**PAI**  
**Orlando, FL**  
**jaeson.h.munro@leidos.com**

### **INTRODUCTION**

Terrain databases for the Modeling and Simulation industry continue to grow in geographic extents, feature densities and content complexities. New training systems are leveraging game engines, requiring growth in database content production capabilities. Unfortunately, the cost of producing these complex 3D cultural models, using manual 3D modeling practices is unsustainable and unaffordable.

The process of assigning 3D model references to vector features in a geospatial database, identified as modelization, has been automated by the Synthetic Environment Core (SE Core) program to significantly reduce database production costs, improve database quality and consistency and increase run-time database performance. SE Core's Automated Modelization Process includes software to analyze the geospatial vector feature data content; identify the unique set of 3D cultural models required – based on feature attributes and the target run-time system constraints; procedurally generate the desired set of 3D cultural models matching the complexity supported by the run-time system; and assign the 3D model reference to the vector features in a geospatial database - including feature placement and angle of orientation. Using automated tools for identification, assignment, and procedural model generation for model creation allows a smaller group of database content creators to focus on a limited number of distinctive models, while generating a dense and complex terrain database.

This paper describes the processes and tools used in the Automated Modelization Process in the production of the SE Core geospatial terrain databases. The paper begins by introducing the SE Core database production architecture, processes and tools. Next the paper introduces the modelization process, and provides details on the Automated Modelization Process and the procedural model generation tools. Next the paper provides details of the first usage on the Automated Modelization Process on the SE Core Integrated Training Environment (ITE) Germany Database. Last the paper ends with an assertion of the benefits of the Automated Modelization Process, and provides insight into the future enhancement planned.

### **SE CORE CVEM STDGC ARCHITECTURE**

The SE Core database production processes are divided into three major phases of production: Master Database (MDB) Population, Database Specialization, and Runtime Database Production. The SE Core CVEM Standard/Rapid Terrain Database Generation Capability (STDGC) architecture diagram supporting these three production phases is shown in Figure 1. The MDB Population phase is supported by the Master Terrain Database Generation Toolkit (MTDGT), the Database Specialization phase is supported by the Dataset Specialization Toolkit (DST), and the Runtime Database Production phase is supported by the Run-time Database Generation Toolkit (RDGT).

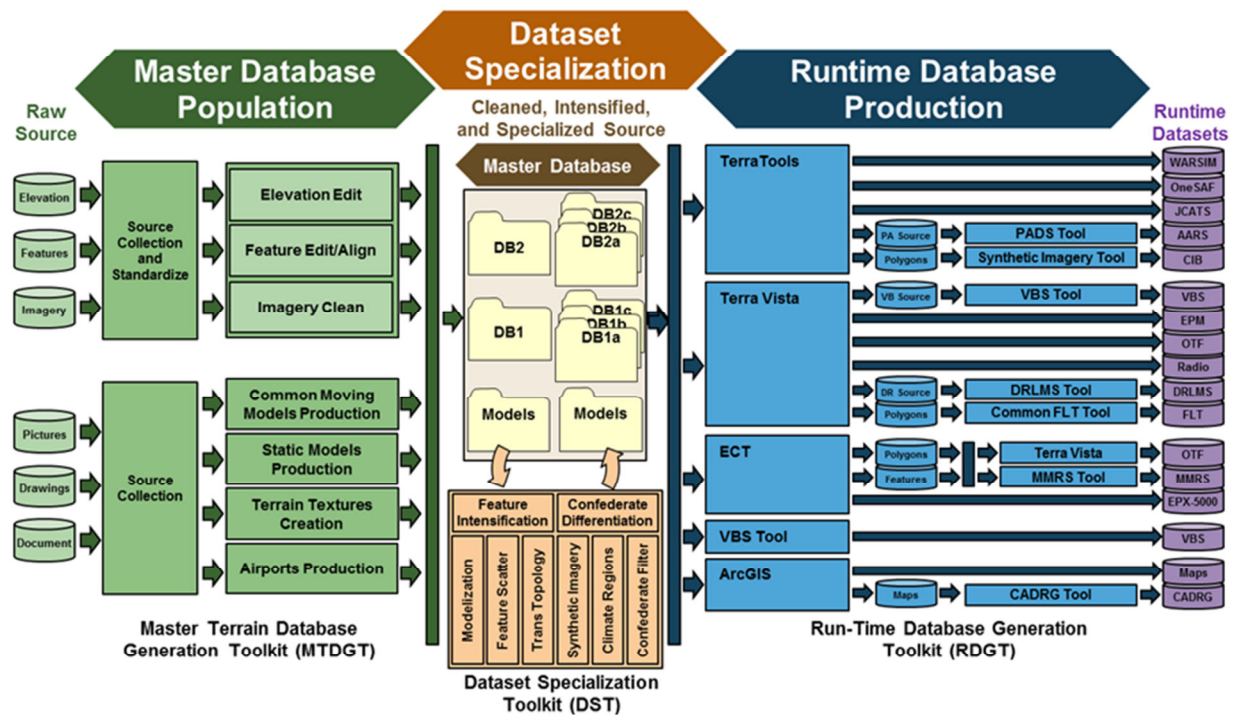


Figure 1. SE Core CVEM STDGC Block Diagram

### MDB Population (Phase One)

The MDB Population phase consists of the collection and standardization of aerial imagery, feature and elevation data; cleaning and aligning of elevation data; and cleaning, aligning and digitizing of the feature data. Also part of the MDB Population is the creation of the moving models, static models, terrain textures and the airfields. At the completion and acceptance of the cleaned data, the Cleaned Source Data is placed in the MDB. The MDB Population architecture leverages commercial off-the-shelf (COTS) products, using ESRI ArcMAP and ArcCatalog for source data collection, standardization and vector feature editing; Intergraph's ERDAS Imagine and Blue Marble's Global Mapper for elevation and imagery editing; Adobe's Photoshop for textures creation; Presagis' Creator and Autodesk's Maya and 3DStudioMax for 3D model development; and Presagis' Airport Wizard for Air Field modeling. The MDB Population phase is focused on capturing an accurate representation of the real world for a specific geographic region, and simulation specific tailoring is deferred to the Dataset Specialization phase.

### MDB Repository

The MDB contains the Cleaned Source Data for all geographic areas, organized by database area and subdivided into data types. The MDB repository is the central component of the STDGC database production process, containing all of the processed content supporting the production of the various run-time terrain databases. The MDB content structure, depicted in Figure 1, contains cleaned source, intensified source, and confederate specialized source.

### Database Specialization (Phase Two)

In order to improve data sharing and accommodate the many run-time formats, the Database Specialization phase was added to the SE Core production process. This phase ensures that data specific to a given use case doesn't pollute the MDB, and provides data differentiation to support efficient use of the data. Database Specialization is divided into two parts, Feature Intensification and Confederate Differentiation, as shown in Figure 2. The Feature intensification includes tools to procedurally generate models, scatter features, and generally intensify the Clean Source data to create the desired simulation content. The Confederate Differentiation part provides fully automated

thinning, filtering, translating, formatting and exporting of the confederate specific content. The Dataset Specialization phase uses government off-the-shelf (GOTS) tools. The intensified source is a fully featured, populated dataset which has the highest fidelity of features. The Intensified Source data is the master dataset for a given geographic region.

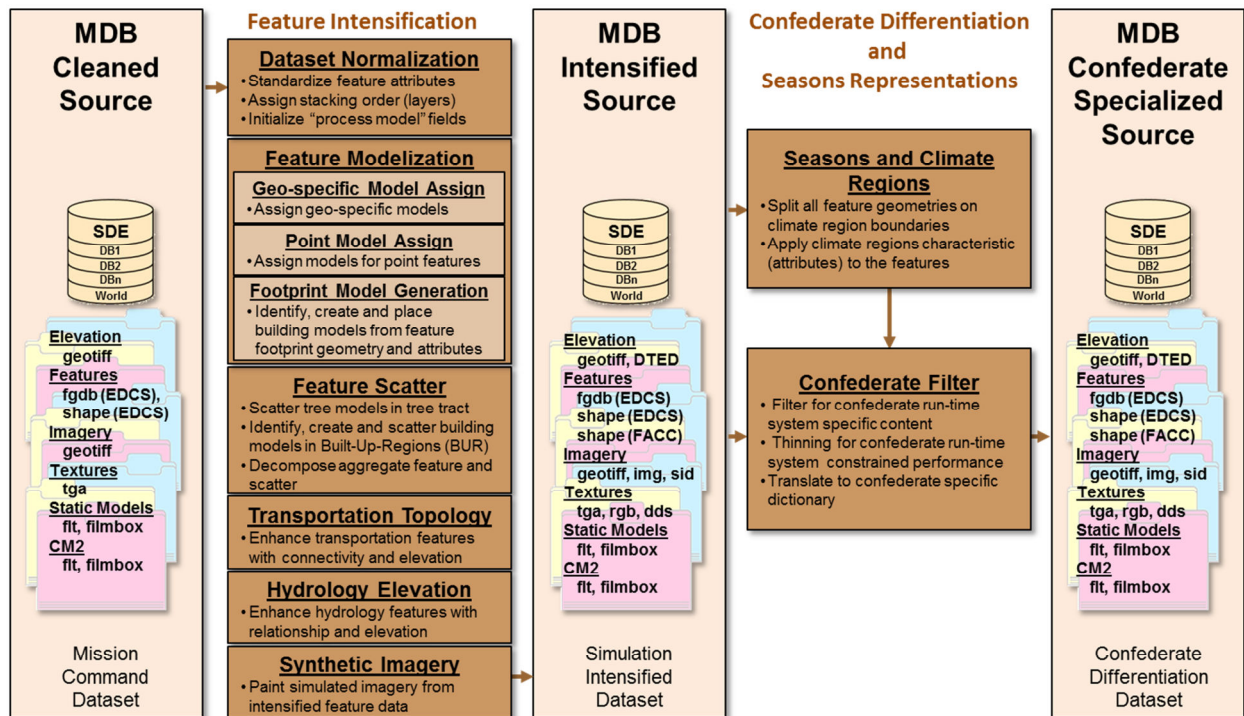


Figure 2. Dataset Specialization Process

### Runtime Database Production (Phase Three)

Once the Dataset Specialization phase is complete, the run-time databases can be produced using the appropriate Database Generation Systems (DBGS). Each of the run-time formats requires specific content and specific formats, many proprietary. The Runtime Database Production phases use DBGSs including Presagis' Terra Vista, TerraSim's TerraTools, Rockwell Collins' Environment Creation Tool (ECT), and Bohemia Interactive Simulations' VBS tools, in addition to others.

### DATABASE SPECIALIZATION (Phase Two)

The modelization process is part of the database specialization process. The database specialization process is composed of adding simulation specific content to the cleaned dataset. As shown in Figure 2, assigning 3D models to specific features in the cleaned dataset is an important step in the dataset intensification process.

### Feature Intensification

Feature intensification includes dataset normalization, feature modelization, feature scatter, transportation topology enhancements, hydrology elevation enhancement, and the generation of synthetic imagery.

Feature modelization is the process of assigning 3D models, by name, to point, line and areal features based on feature classification and attribution. Modelization also includes the generation of cultural feature 3D models using procedural geometry generation capabilities based on building feature footprints, height, roof type and other feature attributes. This Automated Modelization Process provides for the per database tailoring of 3D models based on the dataset feature content and as constrained by the run-time system. (Muller, Wonka, Haegler, Ulmer, & Van Gool 2006)

## **Feature Modelization**

Feature modelization is the process of assigning 3D models, by name, to point, line and area features based on feature classification and attribution. Feature modelization is divided into four stages: 1) Geo-specific feature model assignment, point feature model Assignment, areal feature (footprint) model generation and assignment.

### **Geo-specific Feature Model Assignment**

During the database requirements phase, buildings of significant importance are identified. Geospecific models are created by hand and placed in the specific database model library. These include landmark features used for navigation, or features that are critical to the recognition of a geographic area. Also included are all buildings contained in the training sites that require interior modelling. These are all modelled by hand to include rooms with windows and doors. During vector editing these landmarks or training site features are placed as point or aerial features. During the Geo-specific Feature Model Assignment phase, model names are added to the geo-specific point features, and point features replace areal features.

### **Point Feature Model Assignment**

After the geo-specific models are assigned, all geo-typical point features are assigned geo-typical models from the model library. This includes assigning model such as radio towers, water tanks, churches, parking garages, etc. Selecting geo-typical models requires manually sifting through the model library looking for models that are the style consistent with the geographic area.

### **Areal Feature (Footprint) Model Generation and Assignment**

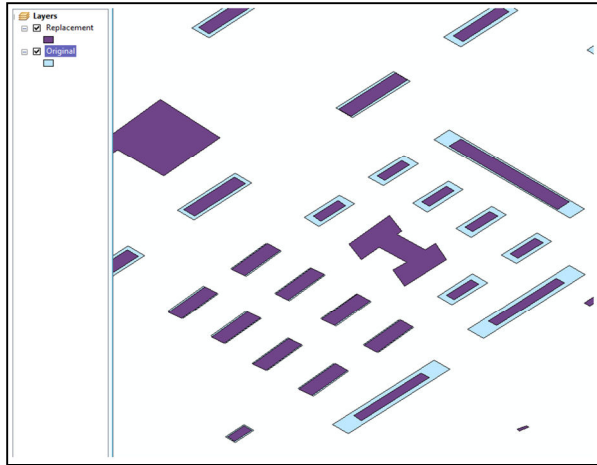
The final process is the new Automated Modelization Process. It includes the generation of cultural feature 3D models using procedural geometry generation capabilities based on building feature footprints, height, roof type and other feature attributes. This Automated Modelization Process provides for the per-database tailoring of 3D models based on the dataset feature content and as constrained by the run-time system.

## **AUTOMATED MODELIZATION**

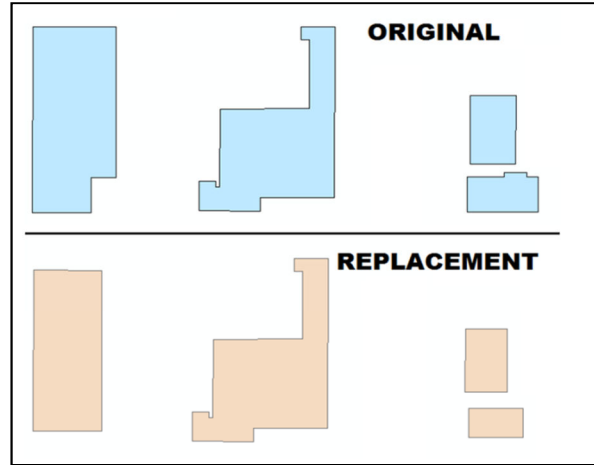
Automated Modelization includes analyzing the geospatial vector feature data content; identifying the unique set of 3D cultural models required; procedurally generating the desired set of 3D cultural models; and assigning the 3D model reference to the vector features in a geospatial database - including feature placement and angle of orientation. The SE Core Automated Modelization Tool leverages the RDECOM-funded Procedural Model Generation Service (PMGS) developed by GameSim.

### **Footprint Analysis**

In the Footprint Analysis phase areal building features in the source file geodatabase are generalized into groups wherein each group can be represented by a common model. Footprints are evaluated based on size, shape, height, and building function. The user may control the degree to which the footprints are generalized. Highly generalized output can be represented by a smaller number of total models, whereas less generalized output is more similar to the source geometry. Figure 3 provides an example of the footprints used as input in the analysis.



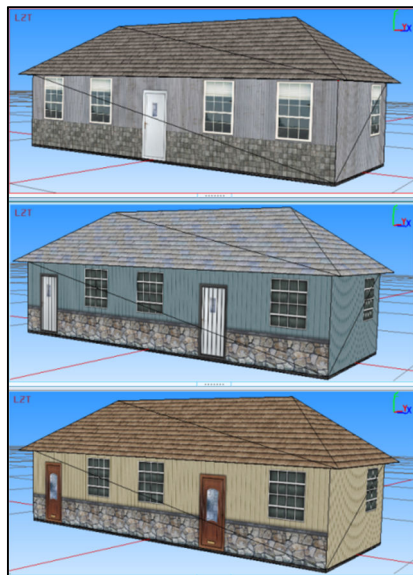
**Figure 3. Original Footprints Overlaid by Replacement Footprints**



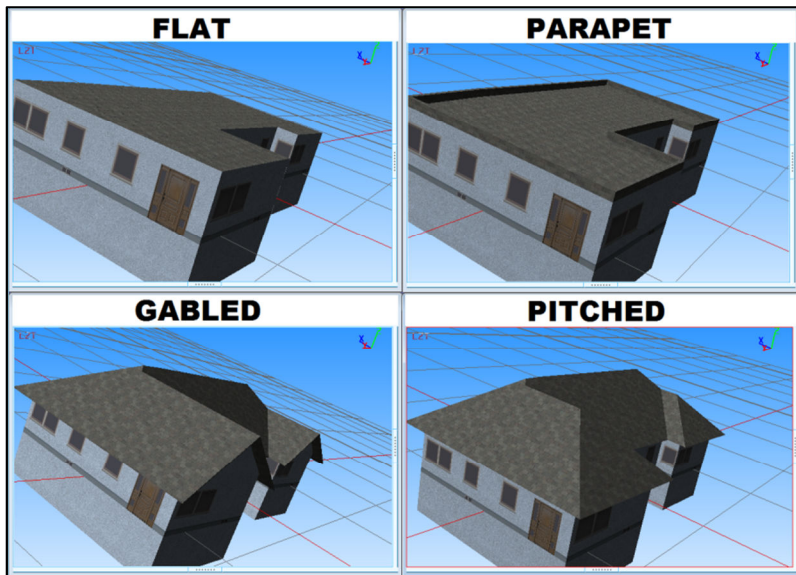
**Figure 4. Feature Generalization Example**

In a typical configuration, some footprints will not be generalized. This occurs when the shape of a footprint is complex, or the size of a footprint is large (see figure 4). The user may also configure the process so that buildings matching a certain criteria are always treated as complex, one-off models. For example a user can decide to always treat stadiums as one-off models, and they will never be generalized.

Resulting footprint groups are assigned one or more sets of textures, and each texture set represents a unique model for that footprint. It is also possible to assign multiple texture sets to a group, and features in that group will be assigned one of these texture sets at random. This is illustrated in Figure 5, where a generalized footprint has been assigned three different models, each with its own texture set.



**Figure 5. Same Model, with Three Different Texture Sets**



**Figure 6. Model Shown with Four Different Roof Types**

The analysis phase packages the geometry of the generalized footprint, along with some model attributes, and a set of textures into a JSON string to be used by the PMGS server during the model generation phase. Some of these attributes include skirt length (basement), roof type, roof slope, and roof overhang. Figure 6 illustrates some of the roof types supported by PMGS.

## Model Generation

During the model generation phase, PMGS generates models which get assigned to building features in the source database. PMGS reads the JSON output from the analyzer, and generates a 3-D model in COLLADA format. The final step in the model generation phase converts this COLLADA model to OpenFlight format.

## Textures

PMGS uses up to seven different component types to apply textures to buildings. A full set of textures for use with PMGS requires one texture for each of the building components (see figure 8).

The filename of each texture indicates how textures are applied to a model. This includes the geographic region, number of floors, horizontal tiling, building component, and horizontal size in pixels.

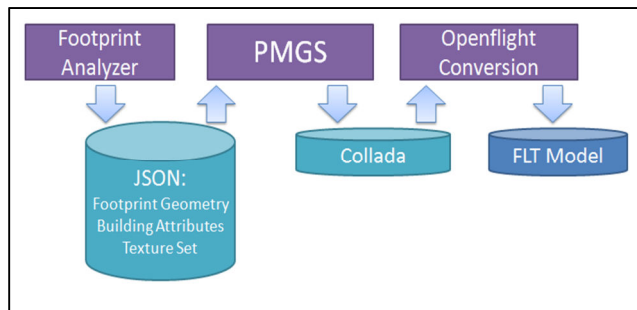


Figure 7. Data Flow through

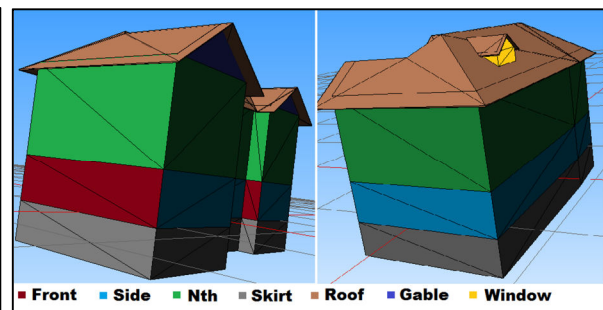


Figure 8. PMGS Building Model Components

## Horizontal Tiling

The tiling defines the way in which a texture can be segmented, truncated and repeated when applied to a polygon. Figure 9 is a front building texture with a horizontal tiling of three.



Figure 9. Texture with Three Horizontal Tiles

A typical application to a polygon will result in some truncation or repeating. In Figure 10, we can see that two of the segments have been repeated to make a better fit to a polygon.

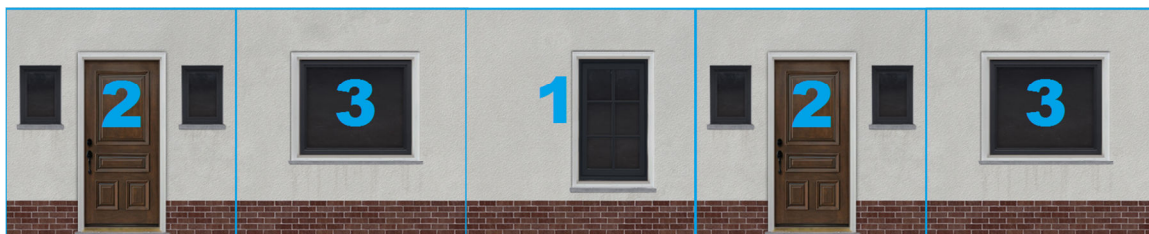


Figure 10. Texture with Three Horizontal Tiles, Two Tiles Repeated.

Applying this tiled texture to the model, it is stretched to make an exact fit (Figure 11).



**Figure 11. Tiled and Stretched Texture on the Front of a Building Model**

### Model Placement

Models are placed in the database by replacing each areal building in the source data with a point feature. This new point feature contains all the attribution of the original footprint areal, along with the name of the model to which it has been assigned, and the angle at which the front face of the model is oriented. The angle of orientation is given in degrees clockwise from north (heading). Each model is oriented to fit the generalized version of the original areal footprint. Generalized footprints will often be symmetrical, or near symmetrical. In this case, there may be more than one orientation that will fit a footprint. When choosing an orientation for a model with multiple possibilities, the model will be placed so that the front side of a building is facing a nearby road (see Figure 12).

Once the new procedurally generated models are placed in the feature database, the “as built” footprints are generated from the new models and are placed in the feature database. This supports producing a correlated footprint output for maps, plan view displays, and constructive simulations that only use footprints (with attributes). The “as built” output footprints, along with the original input footprints are shown in the Figure 3 example.

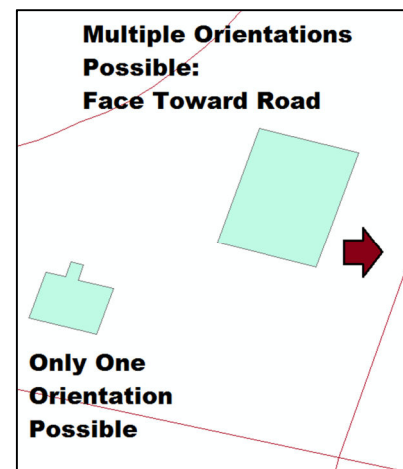
### FIRST USAGE

The SE Core ITE Germany database was selected to be the first usage of the new Automated Modelization Tool. A number of activities were required to make this first usage a success, including creating new textures and verifying the Automated Modelization functionality. It was critical that the Automated Modelization Tool generated the correct mix of instanced models and one-off models.

Instance models are used to represent geo-typical features that share similar footprint size and attributes. A procedurally created model is used multiple times (instanced) in a single database. One-off models are used only once because the combination of footprint size and attributes are so unique that a procedurally created model can only represent one geo-typical feature. The correct mix means that the models produced can be used on the target runtime system, and that the total number of unique models does not exceed the capacity of the target runtime system.

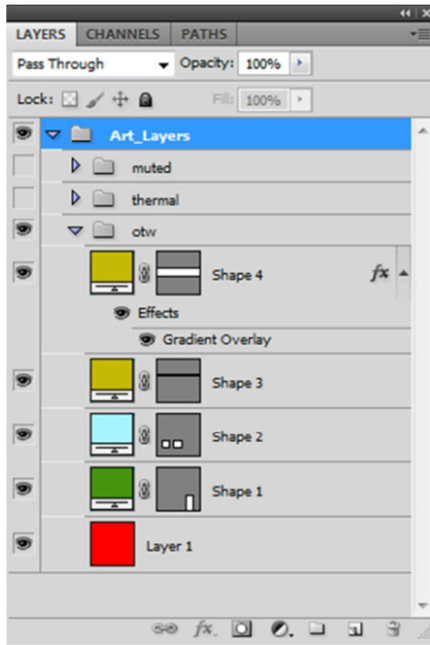
### First Usage - Texture Creation

The first task was to create the new textures required by the procedural model generation tool. A texture creation specification was written to define how the textures would be organized in the Photoshop PSD files. Figure 13 provides an example of the layer definitions for model textures. It includes thermal texture layer, muted textures, and Out-the-Window textures.

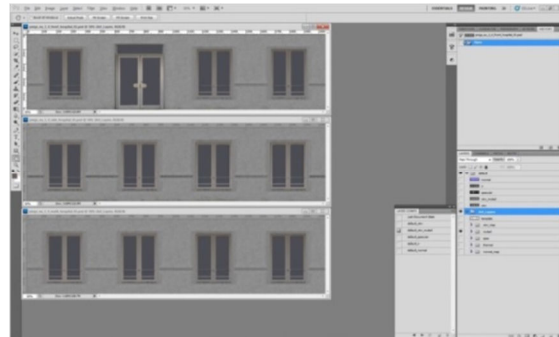


**Figure 12. Two Different Scenarios for Model Orientation**

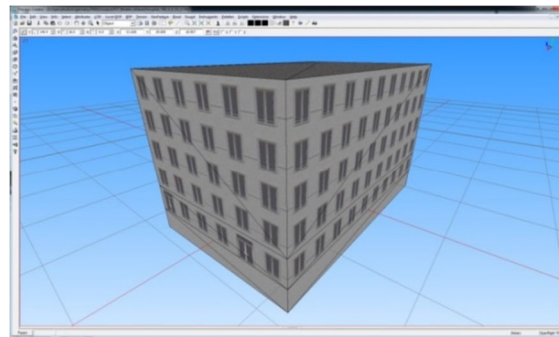
All textures are created to provide efficient use within the procedural generation tools and support efficient processing within the target runtime systems. Figure 14 provides an example of a building texture and Figure 15 provides an example of the resulting procedurally generated building.



**Figure 13. Photoshop Texture Layer Example**



**Figure 14. Example First Usage - Texture**



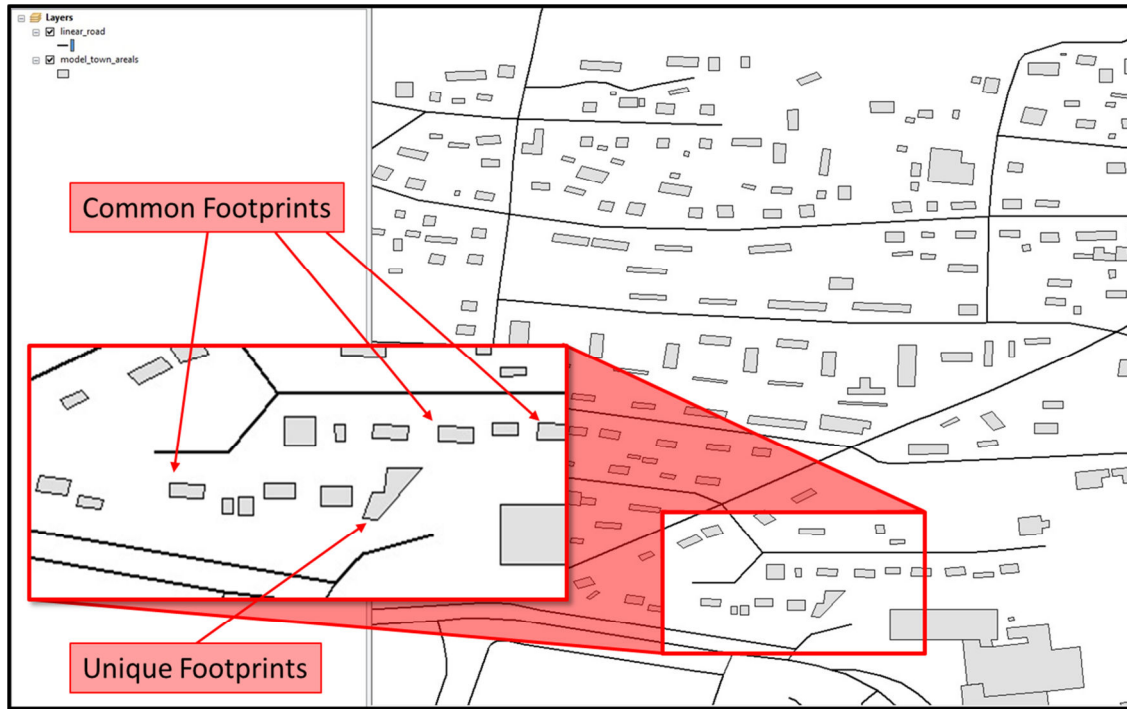
**Figure 15. Example First Usage - Building**

### First Usage - Modelization

Once the database is populated into the MDB, the Database Modelization process begins, and the geo-specific models are hand placed. The geo-specific models represent the landmark features built by hand by the modeling team. In the Germany database all of these models include interiors.

Next all point features are assigned geo-typical models from the model library. This includes placing power pylons, water tanks, etc.

Next the footprint features are analyzed. Figure 16 provides a sample of the vector feature data that was analyzed. The analysis software must be able to “best fit” the footprint regardless of feature orientation and feature extraction artifacts such as geometry skewing.

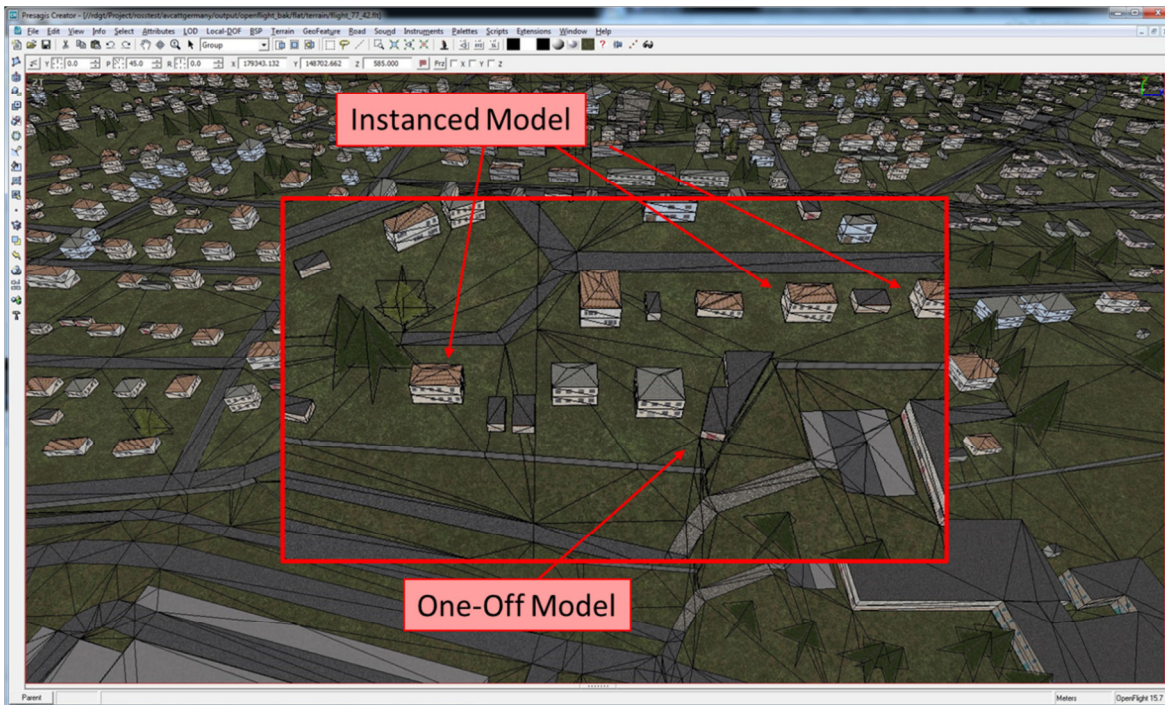


**Figure 16. First Usage – Input Footprints**

Also, note the mix of commonly shaped footprints and the uniquely shaped footprints. The analysis tool must identify the commonly shaped and similarly attributed features and identify a single model to generate.

After the analysis is complete the required models are generated. The Automated Modelization Tool interface provides the capability to select the degree of match between footprints and attributes to create more or less instanced models. This provides the capability to select the total number of unique models base on the capability of the target runtime system.

After the models are generated they are then placed as point features into the vector feature data. Figure 17 provides an example of the placed models.



**Figure 17. First Usage – Model Placement**

### First Usage – Challenges

One major challenge was experienced in the first usage. Footprint skew required adjusting the algorithm. Not all footprints were skewed. The software was enhanced to detect footprint skew and remove the skew before best fit matching could be accomplished.

### First Usage – Results

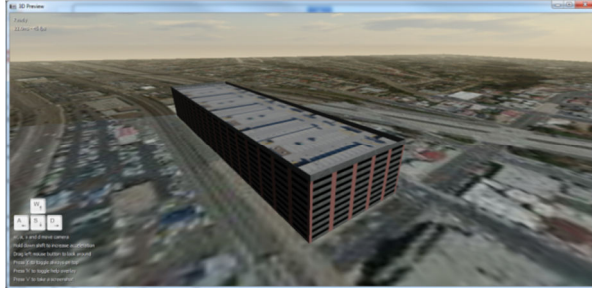
At the final database review the users commented that the Germany database was the most dense and best looking ITE database that SE Core has produced to date. By using similar textures and procedurally generated model geometry the Germany database models looked consistent to the specific geographic region, with the best overall runtime performance of any SE Core database. The Germany dataset had approximately 32,000 areal building footprints in the source. The modelization process generated 4117 unique building models from that source. This included about 300 instanced models, with the remainder being one-off models.

### COST TRADE

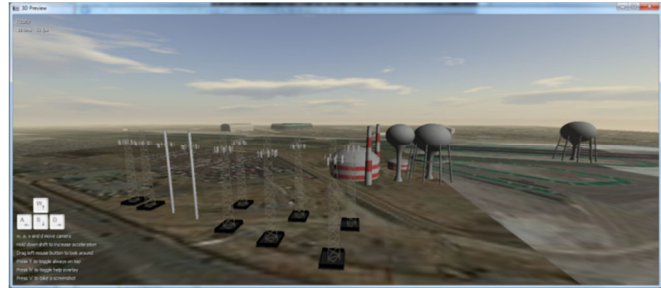
Using automated tools for identification, procedural generation, and assignment of models allows a small group of database content creators to focus on a limited number of distinctive models, while generating a dense and complex terrain database. Before automated model generation was used, models were taken from the SE Core static model library. However, the cost of maintaining the static model library has become unaffordable, and impossible to update to support new image generators and model rendering engines. The SE Core database content creators now focus on creating building with interior models, and producing textures for the procedurally generated models. Automated modelization results in a cost effective approach to generating 3D cultural models that are designed for the specified run-time requirements, saving considerable development time as compared to using commercial 3D modeling tools with manual processes.

## CURRENT AND FUTURE DEVELOPMENT

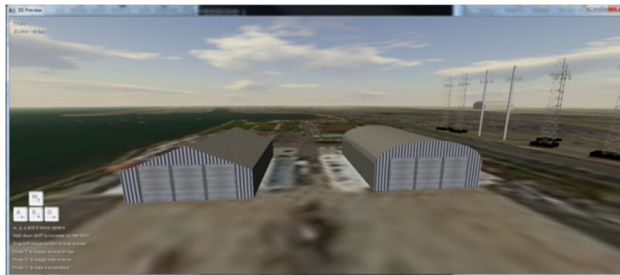
Current automated modelization enhancements focus on supporting the ever increasing demands for faster, cheaper and more realistic databases for training and simulation. This includes enhancing PMGS capabilities as well as tuning the system to increase efficiencies and effectiveness of point model placement.



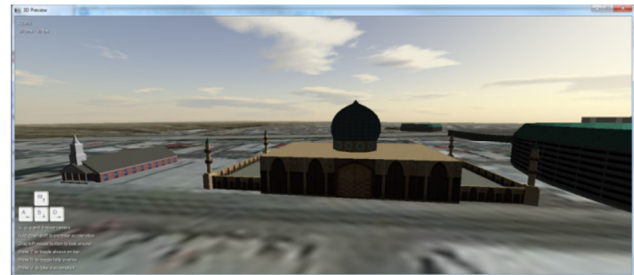
**Figure 18. Parking Garage Example**



**Figure 19. Towers Example**



**Figure 20. Hangers Example**



**Figure 21. Religious Complex Example**

PMGS is being enhanced to procedurally generate new geo-typical feature types such as airport hangers, parking garages, stadiums, athletic fields, airport hangers, storage tanks, water towers, communication towers, churches and mosques. These models will be suitable for ingestion into gaming visual training databases with procedurally generated interiors as required. There are also efforts to increase the procedural generation of building interiors to include interiors with staircases, useful by gaming systems. Figures 18 to 21 provide examples of the new work.

The SE Core Automated Modelization Tool is being developed as a standalone Government Off-the-Shelf (GOTS) tool, and can be requested under U.S. Government Purpose Rights. Also, under the oversight of the PEO-STRI Enterprise Data Services (EDS) project the Automated Modelization Process will eventually be hosted in the cloud and provide web-based on-demand Automated Modelization Service.

## ACKNOWLEDGEMENTS

We acknowledge the contributions of the SE Core team members that were instrumental to the development of the Automated Modelization Tool, including Tony Pelham, Ross Miller, Shawn Shiflett, Robert Pescatore, Pedro Ramos and Randy Toth. Additionally, we wish to acknowledge the vision of Dr. Rob Cox for imagining and funding PMGS tool development at GameSim.

## REFERENCES

- Muller, P., Wonka, P., Haegler, S., Ulmer, A., & Van Gool, L. (2006). Procedural Modeling of Buildings. *ACM Transactions on Graphics (TOG) - Proceedings of ACM SIGGRAPH 2006*, 25, 614-623.