Simulation of Civilian Traffic Rules in Computer Generated Forces

Jon Watkins, Theresa Tamash, Jim Moerk
Dignitas Technologies, LLC
Orlando, Florida
jwatkins@dignitastech.com, ttamash@dignitastech.com, jmoerk@dignitastech.com

Rachel Overman
PEO STRI, SAFAE-ATRI-PSG-EV
Orlando, Florida
rachel.overman@us.army.mil

Julio de la Cruz
US Army Research Laboratory
Simulation and Training Technology Center
Orlando, Florida
julio.delacruz@us.army.mil

ABSTRACT

Low intensity conflict often involves heavy interaction with civilian populations during activities ranging from patrols to convoy operations. As a result, these interactions are important to realistic training. This paper discusses applied research into representations of civilian traffic rules in Army computer generated forces (CGF) applications. While this work emphasizes the needs of Army driver trainers as a primary focus, proper CGF representation of civilian traffic rules are critical to many other training use cases, including convoys, checkpoints, first responder training, and recognition of aberrant behavior in civilians that may be a precursor to an insurgent attack.

This research investigates existing capabilities, deficiencies, and possible future needs of Army entity-level simulations. Two paths were investigated in parallel. One path is more practical and applied, recognizing the current limitations of modeling & simulation source data and system architectures. This path provides functional improvements for near-term transition to programs. In parallel, a more advanced effort is investigating solutions that are more complex and sweeping, therefore complicating practical application. However, this thread illustrates what is possible as a long term solution. This paper discusses this dichotomy of priorities and how these differing goals were addressed.

There are a number of challenges with implementing civilian traffic functionality in multiple phases of modeling and simulation development. Common geospatial sources lack fundamental data needed for accurate traffic simulation, and significant deficiencies in entity-level CGF systems complicate more advanced behaviors. New concepts for path planning and entity avoidance are needed based upon common traffic rules and patterns such as "lane awareness". This paper explores the challenges in current M&S technologies, describes implemented and proposed solutions, and describes how lessons-learned can be applied beyond traffic simulation into other pattern-of-life simulation use cases.

ABOUT THE AUTHORS

Jon Watkins is founder and COO of Dignitas Technologies. He has over two decades of experience with Modeling and Simulation applications, with a particular emphasis on Synthetic Natural Environment representations and services as well as Computer Generated Forces.

Theresa Tamash is a senior software engineer at Dignitas Technologies. She has more than 13 years of experience with simulation training systems, such as CCTT, AVCATT, SE Core, and OneSAF. Her focus has been on Computer Generated Forces development and user interface improvements.

Jim Moerk is a senior software engineer at Dignitas Technologies. He has over 15 years of experience in Modeling and Simulation spanning Live, Virtual, and Constructive applications with a focus in system interoperability and Computer Generated Forces behavior development.

Rachel Overman is a lead engineer at PEO STRI PM GCTT. She has over 15 years of experience with Modeling and Simulation applications, with specific interest in vehicle and environmental behaviors to add the realism to the training environment. Mrs. Overman has provided technical leadership on many Army and Navy programs that include MRAP, Bradley, Bradley Embedded, Abrams Embedded, Common Driver Trainer (CDT), Virtual Convoy Training System (VCTS), and Engagement Skills Trainer (EST).

Mr. Julio de la Cruz is the Science and Technology Manager and Chief Engineer for Synthetic Natural Environment (SNE) and Simulation Technologies. Mr. de la Cruz has over 20 years' experience in government and private sector. As the SNE Chief Engineer for the Science and Technology at the Army Research Laboratory, Human Research and Engineering Directorate, Simulation and Training Technology Center, he oversees the research and development of emerging Database Generation tools and technologies that supports the Command and the Army current and future needs in Modeling and Simulation.

Simulation of Civilian Traffic Rules in Computer Generated Forces

Jon Watkins, Theresa Tamash, Jim Moerk
Dignitas Technologies, LLC
Orlando, Florida
jwatkins@dignitastech.com, ttamash@dignitastech.com, jmoerk@dignitastech.com

Rachel Overman
PEO STRI, SAFAE-ATRI-PSG-EV
Orlando, Florida
rachel.overman@us.army.mil

Julio de la Cruz
US Army Research Laboratory
Simulation and Training Technology Center
Orlando, Florida
julio.delacruz@us.army.mil

INTRODUCTION

This research focus on traffic simulation originated with a call for proposals from the Army Modeling and Simulation Office (AMSO) to investigate ways to extend the use of the Army's OneSAF (Surdu, Parsons, and Tran, 2005) into more modeling and simulation use cases. OneSAF is the Army's premier entity-level computer generated forces (CGF) application, widely used across simulation domains and programs (Hughley, Watkins, Moerk, and Chang, 2006 & Kleinhample and Palomino, 2007). One area of weakness within OneSAF is traffic simulation, and several possibilities exist for improving this area and extending OneSAF's usefulness. Focus was given to the Army's Common Driver Trainer (CDT), which is a virtual trainer used to provide basic skills training to drivers of vehicles such as Stryker and MRAP variants (TPIO-Virtual Training Environment, 2005). Currently, OneSAF does not meet the functional requirements of CDT in terms of computer generated forces (CGF) capabilities, although there is significant opportunity for OneSAF to meet these needs with improvements to traffic capabilities. CDT requires trainees to gain experience in both tactical and civilian travelling modes, thereby requiring surrounding activity with civilian traffic.

While the immediate focus of this research is on OneSAF extensions for CDT needs, there is a much broader potential base for civilian traffic representations. For example, more realistic traffic simulation would support better representations of convoys, road blocks, and movement in low intensity conflicts with civilians in close proximity. In addition, other virtual training programs, such as the Close Combat Tactical Trainer (CCTT) and Aviation Combined Arms Tactical Trainer (AVCATT) could benefit from more realistic representations of civilian movement and convoys.

This paper describes efforts to investigate and mitigate the capability gap between OneSAF and CDT traffic simulation. The effort included two parallel paths of consideration, with near-term effort focused on immediate functional gains in OneSAF capability, and the longer-term thread of experimentation considered improvements that would require more complex updates, potentially spanning beyond OneSAF into areas such as geospatial source data, network protocols, and more. This dual thread required constant tradeoffs between near-term needs and incremental progress versus more sweeping and comprehensive enhancements.

GAP ANALYSIS

Analysis began by eliciting a set of requirements for traffic functionality from CDT. Data was collected on CDT's capabilities as they relate to traffic by observing demonstrations of CDT simulation runs, talking to CDT developers, and analyzing CDT's data inputs. While the data specific to traffic functionality gleaned from CDT developers was limited because the traffic simulation portion of CDT is proprietary, observations showed how simulated entities in CDT behaved and requirements were derived from there. The traffic functionality necessary for CDT training was then compared with what is available in OneSAF. Provided in this section is an overview of the CDT functionality and what is currently available within OneSAF that can be leveraged to meet traffic requirements.

Movement

One difference between entities within CDT and entities within OneSAF is their movement technique. Movement within OneSAF was designed primarily for tactical operations. Civilian entities and behaviors were added later, and they utilize the same tactical movement techniques. In contrast, entities within CDT operate as one would expect civilian vehicles to drive. They travel along roadways and in assigned lanes, avoid obstacles but aim to stay within the confines of the road, obey traffic signals and signs, use turn signals when appropriate, and depending on their aggressiveness settings, can share the road with other vehicles following the same rules.

Table 1: Relevant differences between CDT and OneSAF techniques for entity movement are listed.

Capability	CDT	OneSAF
Stay in lane	Entities stay in their lane.	Entities travel road centerline.
Avoid obstacles	Entities avoid obstacles but try to stay on the road.	Entities avoid obstacles with no consideration of the road's location.
Avoid entities	Entities will stop behind a stationary entity in their lane, or pass a slow moving entity in another lane. Entities appropriately interact with other entities at intersections (take turns, obey signals, etc.)	Entities are avoided with no consideration of the road's location. Entities proceed through intersections, and if needed, will turn wide to avoid other entities.
Path following	Entities stay on their path (which is in their lane) even on tight turns.	Entities often stray from their assigned path due to physical model constraints, and do not stay on the road during tight turns.
Driver Settings	An entity's aggressiveness can be adjusted.	There are no aggressiveness inputs to OneSAF's Driver.

Lanes

OneSAF does have the capability of assigning an entity to travel along a roadway. Routes can be planned loosely along a road on the PVD, and the Control Measure will snap to the roadway features. The movement behavior uses this roadway route to plan an entity's path, which will be directly down the center of the road. The entity itself does not know it is traveling along a road; it merely follows its assigned path. Beyond assigning the entity's initial route, OneSAF does not consider the location of the roadway or associated lanes. Based upon observations, it is clear that CDT entities drive within their lane, change lanes appropriately, and consider their lane assignment and the activity in adjacent lanes when deciding whether to slow or stop behind a vehicle up ahead or drive into another lane to pass.

Because OneSAF's default road following action is to drive down the centerline of a roadway regardless of lanes or roadway directionality (see Figure 1), it is difficult to simulate two-way traffic patterns. At best, an operator could manually place routes over the top of the road, each off-center so as to support a rough two-way traffic pattern. This approach would not only be tedious, but would fall prey to problems with entity avoidance.

There are many secondary challenges derived from lack of lane awareness. For example, dynamic entity avoidance (DEA) algorithms, which are designed to prevent entity-on-entity collisions, cannot respond appropriately by recognizing that on-coming traffic is unlikely to leave their lane. Thus, DEA algorithms indicate that heading changes are needed to avoid a collision, resulting in simulated entities frequently leaving the roadway. Similarly, when traveling on a four lane divided highway and passing slower traffic, preference should be given to staying on the roadway and using the available lane for passing rather than attempting to pass by leaving the roadway.

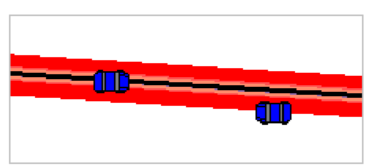

Figure 1 : The vehicles are traveling in opposite directions down a road (along the centerline). One leaves the roadway to avoid a collision with the other.

Traffic Modifiers

CDT represents traffic lights and stop signs at intersections. Traffic lights are synchronized together so east/west roads are green at the same time that north/south roads are red. Entities obey the traffic lights, as well as the stop signs. There are also yield signs in some areas, and entities obey these as well. Entities will also obey the posted speed limits within CDT. OneSAF offers some limited representations for things like speed limits on road features, but the representations are incomplete and are not used by the simulation software. Similarly, some method would be required to allow OneSAF's state for stoplights to match the visual representation in a system like CDT.

Traffic Cloud

In a CDT training run, manned simulators (called the ownship) provide the training environment for soldiers. Depending on the training goals, additional traffic on the roads surrounding the ownship is needed. When an ownship gets near a roadway in a CDT training run, it automatically becomes populated with entities moving in the vicinity of the trainee. There are options available so the instructor can specify the density of entities in the area and the aggressiveness of the entities. Entities are created so the specified density is maintained in the vicinity of the ownship. New entities pop up in the direction of travel of the ownship so that there is always a flow of entities. Similarly, entities that are moving away from the ownship are eventually removed from the simulation. The effect is that there is activity in the vicinity of trainees as they are driving around, but the simulation need not maintain thousands upon thousands of entities fully populating all areas of the database.

OneSAF has crowd behaviors that act similarly to a CDT Traffic Cloud, but for people and not vehicles. A group of people will randomly move around a specified area to create more realistic urban environments. OneSAF also has a Travel Roads Randomly behavior that can be used to order a vehicle to travel along the road network, randomly choosing a direction at each intersection.

CHALLENGES

Analysis of traffic functionality requirements and OneSAF capabilities showed that there is limited functionality within OneSAF that can be leveraged for short term improvements. Several challenges also became apparent to achieving the complete, long term solution. Discussion follows on the over-arching challenges associated with providing a full set of traffic features within OneSAF that is sufficient to meet the needs of a trainer such as CDT.

Data Representation

One of the initial challenges was representing the data needed to properly model traffic within a simulation. Analysis showed that some traffic related data is represented in OneSAF, but not completely. For example, each road feature has a data attribute (VEHICLE_TRAFFIC_FLOW) that describes the directionality of the road (two way or one way). When it is set to 'one way' there is no indication of the direction traffic should flow. Marking it as one way is probably sufficient to representing the road correctly on visuals with no double or single yellow line, but knowing the direction of the traffic flow on the road segment is vital to a roadway being navigated by entities. Another example of incomplete representation is that for lanes. Each roadway has an attribute to specify how many lanes a road has (PATH_COUNT), but there is no information on how many lanes go in each direction (e.g. if there are an odd number), if any lanes are turn lanes only, or how wide the lanes are.

In several cases, no meaningful data structures are available to represent traffic related information in OneSAF. For example, there are limited structures to represent traffic lights and stop signs, but these are insufficient to meet modeling needs within OneSAF. In addition, there is not enough information to model intersections fully, such as a complete representation of each lane's location, as well as the valid connecting lanes, and all associated signals and their relationships to each other and the lanes.

OpenDRIVE provides a complete data model that can represent the data needed for traffic modeling (Dupuis, Strobl, and Grezlikowski, 2010). The OpenDRIVE data model was designed for representing roadways, their connections, and all data needed for traffic modeling. A data model such as OpenDRIVE may provide a solid long term solution, but integrating it into OneSAF is too involved and lengthy to provide short term gains.

Incomplete Database Content

Although available data representations are a significant concern, there are instances where OneSAF's Environment Data Model (EDM) provides for a lot of potential attribution, but the terrain database is generally not well populated by database producers, and not all aspects of the EDM are utilized by the run-time software. So, for example, while the OneSAF terrain database can denote whether a road is one way or not, this value is rarely filled in properly. Similarly, speed limit is supported as an attribute for roads, but again is generally defaulted. Experimentation for this effort was facilitated by a select set of terrain databases that had solid content for some of these attributes.

Source Data

Even if complete data representations and perfect database generation toolsets to populate those representations were available, often source data to provide the foundation for traffic information is simply lacking. Common modeling and simulation geospatial data sets often lack attribution identifying road directionality, location of traffic signals, indications of lanes (e.g. dedicated right turn lane), speed limits, and so on. Clearly, much of this information is available in modern in-vehicle navigation systems, but it is generally the case that M&S database generation processes will not correlate with this type of data as an input. This is a challenge both in using the current OneSAF EDM and if the OpenDRIVE data model is leveraged in the future.

Traffic Mode

OneSAF is designed to be the semi-automated forces system for many training systems – CCTT, AVCATT, Combined Arms Command and Control Training Upgrade System (CACCTUS), among others – most of whom do not need entities to follow traffic rules. Entities in OneSAF were designed initially to travel tactically, traveling across various terrain types, ignoring roads, moving around other entities as needed, etc. One particular challenge with adding the capability to OneSAF for entities to travel in a "Traffic Mode" where they travel on roads and follow traffic rules is choosing an efficient and easy to use way for the user to specify which entities should travel in traffic mode. For example, it would be inappropriate for a fighting vehicle to stop at a stop sign while reacting to an air attack, or when assaulting an enemy position.

Currently, OneSAF has two behaviors that an operator can assign to entities and they will travel along the centerline of the road network. These entities use the same driver and mobility models as entities that are traveling tactically. As will be discussed in more detail in a later section, modified driver and mobility models are required in order for entities to achieve a true adherence to traffic rules. Since entity types are composed of their models, this presents a challenge. Traditionally, if a new or modified model is required, a new entity type is created. This would require a new set of civilian entities within OneSAF, who would only be able to travel according to traffic rules. This would not only make the baseline larger (more files) but would create a maintenance issue down the line. Ideally, existing entity compositions could be used, with some sort of "switch" to be in traffic mode.

Needed Changes Span Architectural Layers

The needed changes within OneSAF itself are not easily isolated, and affect many different layers of the codebase. Since short term improvements within a tight timeframe and budget are an important objective, this presents a challenge. Isolated changes that affect small parts of the code don't require as much cooperation from the OneSAF team in design and code phases and integration testing doesn't take as long. In order to achieve even the smallest of traffic following functionality, modifications have to be made to the Environment Runtime Component (ERC), which processes terrain data, driver components, mobility models, behaviors, and common services. When considering a full system context beyond just OneSAF, additional issues arise, including the visual representation and distribution of data such as the state of stop lights, brake lights, turn signals, etc.

EXPERIMENTATION

The Gap Analysis provided a good understanding of the requirements of a Semi-Automated Forces (SAF) system to provide traffic functionality within a virtual driver trainer. Experimentation was performed within the OneSAF baseline to determine where progress could be made in the short term, and to see what larger scale changes might be needed for a long term solution. For each thread of functionality needed, results are outlined below of experiments, challenges discovered, and ideas for both short and long term solutions.

Lane Awareness

Entities following traffic rules must be aware of both road and lane boundaries. They need to travel within a lane, appropriately leaving their lane only to pass, turn, or change lanes. This is a complex problem, but via experimentation short term gains were achieved by starting with the simplest piece and building on it. At the most basic level a determination was made of how many lanes are on a road and where they are located so the entity can travel in one of the lanes instead of always driving down the centerline of a road.

Upon investigation it was discovered that OneSAF's terrain database had useful attribution for extracting lane information. Road segments have a WIDTH (in meters) and a PATH_COUNT attribute that were used to calculate the number of lanes on a road. The data in the terrain database is not always complete, so when needed, values were defaulted to a simple two lane road. The VEHICLE_TRAFFIC_FLOW attribute was used to determine if a road is one way or two ways, and the direction of travel for one way roads was calculated based upon point order. Fortunately, databases generated by the Synthetic Environment Core (SE Core) program (Dukstein, Watkins, and Deakins, 2007) have these attributes and others filled in accurately for roads, including correlation with the roadway textures in the visual database. Thus, SE Core databases were leveraged for experimentation so as to minimize defaulted attributes.

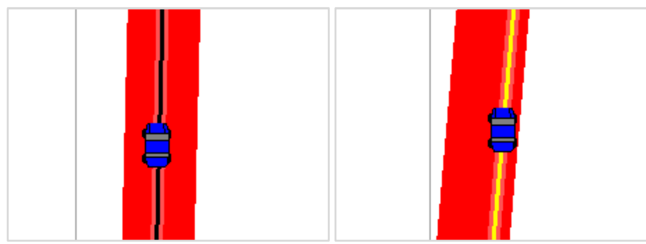

Figure 2: Before, on the left, a OneSAF vehicle is traveling down a route in the center of the road. After, the vehicle on the right is traveling in the correct lane.

In initial experiments, a representation of the lane is not stored and does not persist. Instead a utility was developed to move a road route from the centerline of the road into the center of the requested lane based upon the traffic flow (one way vs. two way), overall road width, and path count. After calculations were completed, the notion of the lane was lost. As a result, entities traveled generally within their lane, although some issues were encountered with OneSAF entities following the ordered route accurately. Of course, if the entity encountered an obstacle or an entity traveling in another lane, or had a hard time making a corner, he would make no effort to remain in his lane (or on the road) or to quickly return to his lane. It was also discovered through experimentation that there was no way to reliably determine the direction of travel for a one way road segment with the data available. However, the road segment point order was typically consistent with the direction of travel and was sufficient for the small scenarios required for the experimentation.

Certainly, an entity traveling within a lane is a step in the right direction, but much more is needed. In order for the entity to remain in the lane during all phases of travel, the road width and lane information must persist within the simulation and be available to many mobility services - dynamic entity avoidance (DEA), obstacle avoidance (OA), path planning, collision recovery, and route generation. As but a single example, DEA needs lane data so that entities can draw conclusions about when to avoid other entities around them. For example, an entity in your lane of motion must be avoided if it is moving slower or is stationary. However, an entity that is driving in a different lane need not be avoided, even if projecting paths would indicate a collision is imminent, such as head-on traffic in two different lanes both approaching the same turn in a road. By default, OneSAF's DEA algorithms are very conservative and react any time entities get within even rough proximity, causing entities in lanes to frequently change heading to avoid collisions that would never happen as long as all parties stayed in their lane. Unfortunately, this requires a complex algorithm to determine if other entities are driving in a lane or not so they can be reasoned

Figure 3: A vehicle traveling in a lane as seen on the on Dignitas' stealth viewer.

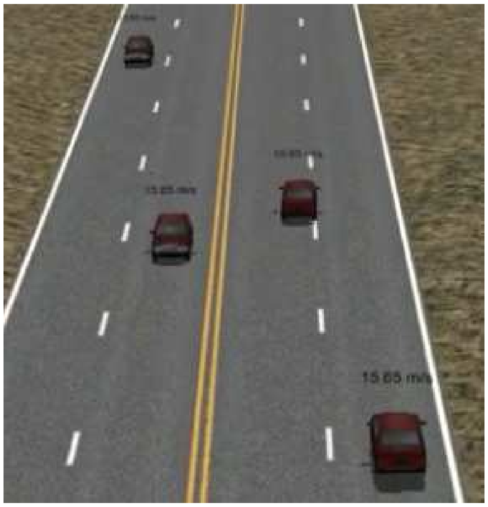

Figure 4 : Several vehicles traveling within lanes. The third vehicle from the left still altered his path some towards the dotted lines to avoid the vehicle in the opposite lane.

on relative to the planning entity's lane. This will be challenging to achieve partly because of the many layers of the architecture that need the lane information, but also because the current tactical travel mode of being lane un-aware must be maintained for fighting vehicles.

Another challenge is determining an efficient way, both in terms of calculation time and storage space, to represent a lane. Long term, the solution to this could be a representation like OpenDRIVE. Short term, lanes must be calculated based on attribution within the terrain database either on the fly as needed, perhaps with a cache system, or store the lane information for all roads on the terrain.

Path Following

Once an entity is traveling within a lane, it is important for the entity to "stick" to its assigned lane by closely following its assigned path. It might seem that this would fall under the "Lane Awareness" thread discussed above, but the OneSAF movement code does not have knowledge of the entity's assigned path. This code is based on the physical capabilities of the entity's vehicle class. The code receives a requested velocity and based on the entity's current state, orders the entity to adjust its velocity based on its physical capabilities.

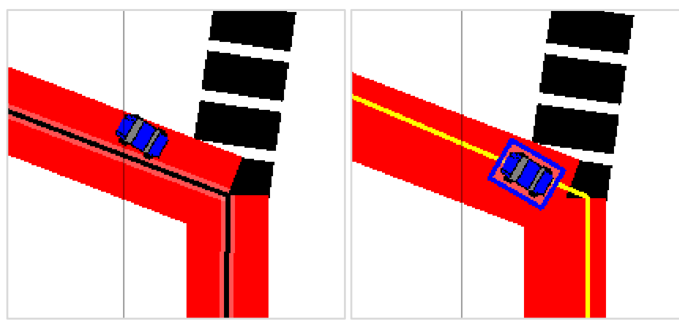

Figure 5: On the left, before any improvements, a OneSAF vehicle just drove around the corner. The black line is the entity's assigned route, down the center of the road. On the right, the vehicle just rounded the corner along his lane route (the yellow line) with path following improvements.

There are several reasons that entities within OneSAF don't stay precisely on their assigned path, but few matter to the existing OneSAF use case. Entities generally follow their paths, but don't stick precisely to their path as is needed for traffic entities driving in narrow lanes. In fact, OneSAF current default behavior of driving down road centerlines minimizes the issues arising from failure to say on a path. For example, an entity making a right turn has the full width of the entire multi-lane roadway to address the turn. Similarly, a vehicle crossing an overpass down the middle is less likely to collide with the overpass rails. As noted before, dynamic entity avoidance is already very defensive, causing entities to shy away from each other, thus mitigating the need to stick closely to assigned paths through narrow lanes.

However, for realistic two-way traffic, entities need a "ride on rails" style of movement that treats staying on their assigned path as a priority over physical capability such as turn radius. To achieve some short term gains in improving an entity's adherence to their path and to learn about possibilities for a long term solution, analysis and experimentation was done with the existing mobility models. Low resolution mobility models were tested, examining the hypothesis that they ignore some of the physical constraints. They do, but to an extent that has the entity jumping around and looking quite jerky. In investigating the medium resolution mobility model, a few parts of the algorithms were discovered that could be adjusted to keep the entity on his assigned path better. For example, the max speed an entity was permitted to travel while cornering was decreased. In addition, entities were allowed more application of the brake, which slowed the entity down on corners even more. Going slower on corners kept the entity on his assigned path better (see Figure 5).

The small improvements made allow some short term gains in keeping an entity on his path in the absence of other entities and obstacles. Based upon experimentation and previous experience with movement control algorithms, it is clear that tinkering with the existing mobility models will not be sufficient to meet strict lane travel needs. For example, vehicles still stray from their route and their assigned path even in the simplest of circumstances, (see Figure 6). This example shows a vehicle straying from his path on a straight portion of road with no obstacles. Instead, a new mobility model is possibly needed that will adjust the priorities of the model, so that it considers staying on the path its top priority. The model would still need to consider the physical limitations of the entity but would stretch its capabilities when appropriate. This approach should ensure the entity's movement is still realistic looking.

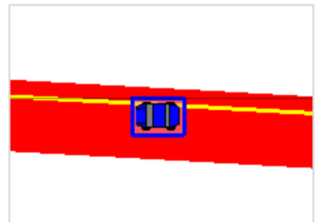

Figure 6 : A vehicle straying from assigned lane route on a straight section of road with no obstacles.

Traffic Modifiers

Once entities are traveling precisely on their path, within the appropriate lane on the roadway, order needs to be maintained among them. In the real world, and in simulation, this is done with road signs and signals. In this case, the needed features are speed limit signs, stop signs or traffic signals at intersections, and yield signs. The biggest challenge with implementing traffic modifiers is a lack of data. The existing terrain databases, including the ones being using from SE Core, have limited data for speed limits and no data for coordination at intersections.

In the first round of experimentation, data leveraged for the speed limits was available on each road segment via the VEHICULAR_SPEED_LIMIT attribute. When the limit was clearly wrong or missing, a reasonable speed was used as a default. The two behaviors (directed travel and random travel) used different approaches to looking up and applying the speed limit, but both approaches achieved the limited speed during entity travel. With only slight adjustments, one or both of these approaches will be effective in both the short and long term.

The biggest challenge in terms of adding support for intersection coordination is representing the traffic signals. There is no data representation in OneSAF's terrain database that can be leveraged for topology or state, nor a representation of signals that can be seen on the PVD. Similarly, there is no extant method to allow light states to be correlated between OneSAF entities and a visual representation. To achieve a short term solution for experimentation, a dynamic terrain feature, the Dragons Teeth, was used to represent traffic lights. Dragons Teeth can be placed on the terrain during scenario setup or during a simulation run in locations where traffic lights should be located. This feature has the added advantage that it is supported as a dynamic feature through OneSAF's V-DIS implementation, thus providing a means to place this object and have it appear on a visual display. An attribute on the Dragons Teeth was adjusted to represent the red, green, or yellow state of the light, and the Dignitas' Veritas Viewer stealth product was updated to display Dragons Teeth as a visual traffic signal with light states so the signals

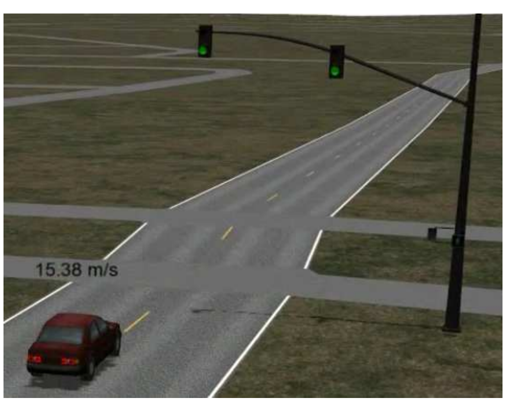

Figure 7 : A green light (represented by Dragons Teeth internally) and a vehicle moving through the intersection (speed displayed above vehicle).

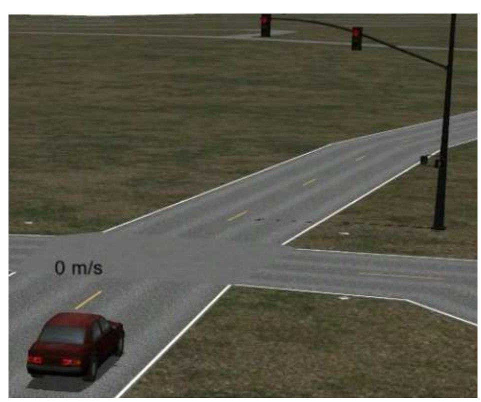

Figure 8 : A red light (represented by Dragons Teeth internally) and a vehicle stopped prior to the intersection (speed displayed above vehicle).

could be seen in a virtual environment (Figure 7 and Figure 8). Within OneSAF, the travel behavior monitored the state of the next signal on the entity's route, and when in range, told the entity to stop upon a red or yellow, and continue with a green signal state. This experiment was successful as a method to illustrate what could be done. Traffic lights were visible on both the visual system and on the PVD and entities appropriately stopped depending on the signal's state. In addition to being able to demonstrate working traffic signals in OneSAF, implementing this approach gave light to what a long term solution entails. A long term solution requires an appropriate feature representation along with decisions as to whether traffic signals should be a predefined part of terrain databases, a dynamically placed object, or both. Either approach provides unique challenges, since traffic lights must be associated with connected features (i.e. which lanes are controlled by which signals) and light states must be managed in a way that results in logical controls, such as orthogonal signals never having green at the same time.

Since there is no existing data or data representation for traffic signals, a long term solution involving OpenDRIVE makes a lot of sense. OpenDRIVE was designed to represent roadways and how the different parts of the roadway system should interact. Roads, lanes, sidewalks, intersections, signals, etc. are all linked together so a simulation can understand their relationships. Since OpenDRIVE is actively used in other simulations (mostly for commercial automotive testing), there is work being done to generate tools and datasets for easing the burden of generating the needed data in OpenDRIVE format (Bentley, K.). Without adopting a solution involving a data representation like OpenDRIVE, users will have to manually place traffic modifiers on the terrain. Scenario generation techniques can be leveraged so this does not have to be done for each scenario, but nonetheless, it seems more work than necessary.

Traffic Cloud

Previous sections of this document describe improvements made to support basic road traffic capabilities at the individual entity level within OneSAF. While these improvements provide a significant benefit on their own, they are not sufficient to support the volume of traffic clutter necessary for driver trainer simulation systems like CDT. It would be too cumbersome for an operator to effectively support the required level of traffic clutter using existing capabilities in OneSAF (a tedious cycle of creating entities, assigning behaviors, and removing entities).

A new capability was added to OneSAF to allow the SAF Operator to define and manage traffic clutter around an ownship entity. The new Traffic Cloud capability allows the user to specify an ownship entity, a traffic entity spawn location, a traffic cloud radius, and a traffic entity spawn rate. Once activated, the Traffic Cloud automatically creates civilian traffic vehicles at the spawn location at the specified spawn rate. Each new traffic vehicle is randomly chosen from a predefined set of civilian entity types to provide a diverse traffic pattern. Each traffic vehicle is assigned the modified Travel Roads Randomly behavior immediately upon creation and travels randomly on the road network to simulate a traffic pattern around the ownship. The ownship entity location is monitored and serves as the center of the Traffic Cloud. Traffic entities are removed from the simulation once they are outside of the radius of the Traffic Cloud. The Traffic Cloud capability along with other traffic improvements (lane travel, path following, traffic lights) achieves a big step towards realistic traffic simulation within OneSAF.

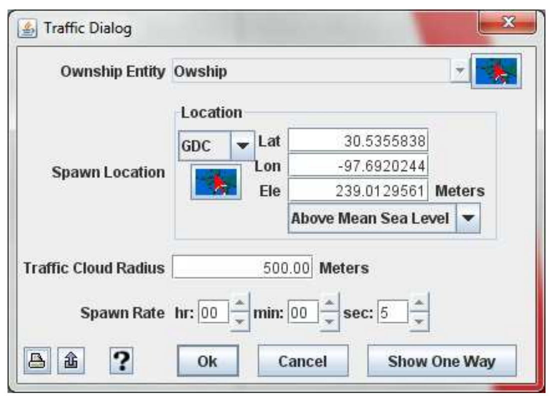

Figure 9: The Traffic Cloud configuration dialog.

The functionality of the Traffic Cloud will be built upon as work continues towards meeting the full requirements needed for CDT. In the future, the user will be able to adjust more of the characteristics of the Traffic Cloud including entity density and driver aggressiveness. Since many Traffic Cloud characteristics are likely to be similar for a particular scenario, the Traffic Cloud configuration will be able to be saved as part of a OneSAF scenario. This initial capability was developed not only as a means to provide an initial traffic pattern simulation, but to also provide a framework for more advanced experimentation. The Traffic Cloud framework allows for experimentation with different traffic patterns. The initial implementation supports only random road movement for individual vehicles. More sophisticated behavior models can be integrated within the Traffic Cloud to provide more realistic traffic patterns. For example, a rush hour model could be developed to generate high traffic flow from residential areas to commercial areas in the morning, and vice versa in the late afternoon. These new models could replace or be combined with the existing random movement behavior in the future to provide a more diverse traffic pattern.

A significant challenge for the Traffic Cloud is identifying the best approach for traffic vehicle creation and deletion from a performance standpoint. OneSAF bases many of its performance benchmarks off of "entity count", so based on how many entities are in a scenario, the user knows how many simcores are needed (the backend workhorse of the simulation). With many entities being created on the fly for the Traffic Cloud, it presents a challenge to keep performance measures accurate. Several options are apparent, such as pre-creating the Traffic Cloud entities during scenario creation time, when the Traffic Cloud is configured. These entities would be in an "entity pool" where they would be hidden from view until placed on the roads and ordered to travel. When they reach the edge of the area, instead of being deleted, they will be hidden again and available for re-use. In addition to OneSAF performance implications, it is also necessary to understand the performance characteristics of all simulators and visual systems within the virtual training environment. The Traffic Cloud framework will allow swapping out which method is used during experimentation so the best approach for all affected applications can be identified.

COLLABORATION

Developed capabilities aim to leverage related work in industry and benefit Army M&S programs beyond OneSAF.

Synthetic Environment (SE) Core Common Virtual Environment (CVE) program

Where the primary focus has been on addressing run-time representation and functionality issues, the SE Core CVE program has been executing an effort focused on improving transportation networks in database generation (Pivonka, Johnson, and Bentley, 2013). These efforts thus dovetailed nicely, allowing collaboration across efforts and allowing each effort to focus in the area where it is strongest. The CVE program provided the most complete attribution found among OTF databases, allowing the experimentation effort to progress faster than initially expected. CVE OTFs were particularly strong with lane information (correlated with visual lanes) and speed limit definitions. Although one-way roads were clearly marked in CVE-built OTFs, this data could not be used effectively because there is no convention to establish which way the road is supposed to go. The two efforts exchanged information of mutual benefit, while helping to align the long-term objectives for traffic modeling.

SPAWN Small Business Innovation Research (SBIR) Project

This effort was supported by Cognitics developers working a research effort related to complex transportation network representations. Elements of their work and its application to SE Core are captured in (Bentley, K.). This complementary work is looking at techniques to represent complex traffic data such as cloverleaf overpasses and taper lanes (e.g. turn lanes). In addition, this effort is looking at techniques to allow OpenDRIVE data to correlate with modeling and simulation databases, thus providing a complete set of visual database content, OneSAF database content, and complex topological and geometry data in OpenDRIVE data structures. While the advanced representation could not be leveraged directly, it provides a view to a longer-term, more complete solution.

Common Driver Trainer

The primary focus of efforts described herein is the CDT program itself, namely meeting CDT requirements through a common OneSAF solution that can benefit all of the M&S community versus a proprietary solution. The work described in this paper will continue forward as part of the CDT follow-on program implementation.

CONCLUSION

The primary goal was mitigating the capability gap between OneSAF and CDT traffic capabilities. A series of capabilities is being integrated into the OneSAF baseline, starting with basic lane awareness and selected movement behaviors. Much of the capabilities described in this paper including lane awareness, improved path following, traffic modifiers and the traffic cloud will be directly transitioned into the CDT program as OneSAF is integrated as the primary civilian traffic capability. Because OneSAF is widely used throughout Army programs, the output of this effort will provide direct benefit to a wide range of Army programs that require higher fidelity urban traffic simulation than what is currently available in OneSAF.

While this research has clearly provided short-term benefits to the Army, there is still a significant opportunity to develop more realistic traffic capabilities to support Army training needs. Further effort is required to address the longer-term and more overarching areas described in the experimentation section. This includes the adoption of a standard data model, such as the OpenDRIVE specification, for both offline and runtime representation of relevant traffic information. While this is not necessarily a technically daunting task, it requires heavy coordination between various stakeholders within the Army simulation community. This research effort has helped to initiate, and can help guide, collaboration between community stakeholders necessary to address database generation in support of long term improvements to traffic simulation for the modeling and simulation community at large.

ACKNOWLEDGEMENTS

The authors would like to thank COL Evans of the Army Modeling and Simulation office for his sponsorship of this work and the SE Core CVE program for their ongoing collaboration. In addition, we appreciate OneSAF's support of our work, including integration of capabilities into the core baseline.

REFERENCES

Bentley, K. *Modeling the Real World - Procedural model generation for training and mission rehearsal*. Retrieved June 6, 2013 from http://www.cognitics.net/sites/default/files/Modeling%20the%20Real%20World.pdf

Ceranowicz, A., & Torpey, M. (2005). Adapting to Urban Warfare. *JDMS, Volume 2, Issue 1, pages 3-15*.

Dukstein, G., Watkins, J., & Deakins, R. (2007) Integration of OneSAF Environment Runtime Component into Existing Virtual Simulations. *I/ITSEC 2007 Proceedings*.

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.

Goerger, N.C., Teague, E., LTC Simon, Goerger, R., Griffin, G., & Richmond, P. (2007) Shaping Insurgent Route Selection Using Traffic Flow Strategies. *I/ITSEC 2007 Proceedings*.

Gratch, J., & Marsella, S. (2003). Fight the Way You Train: The Role and Limits of Emotions in Training for Combat. *The Brown Journal of World Affairs, Volume X, Issue 1, pages 63-75*.

Hughley, P., Watkins, J., Moerk, J., & Chang, A. (2006). Integration of OneSAF Objective System into Existing Virtual Programs. *I/ITSEC 2006 Proceedings*.

Kleinhample, R., & Palomino, J. (2007). Adopting OneSAF as the Core Entity Driver for BLCSE. *I/ITSEC 2007 Proceedings*.

Pivonka, J., Johnson, M., & Bentley, K. (2013). Simulation of Civilian Traffic Rules in Computer Generated Forces. *I/ITSEC 2013 Proceedings*.

Surdu, J.R., Parsons, D., & Tran, O. (2005). The Three Block War in OneSAF. *I/ITSEC 2005 Proceedings*.

TPIO-Virtual Training Environment. (2005). *Initial Capabilities Document for Common Driver Trainer (CDT)*. Retrieved May 24, 2013 from
http://www.acqnotes.com/Attachments/ICD%20for%20Common%20Driver%20Trainer.pdf