VERIFICATION AND VALIDATION (V & V) OF FEDERATION SYNTHETIC NATURAL ENVIRONMENTS

Robert F. Richbourg, Robert J. Graebener, Tim Stone, & Keith Green
Institute for Defense Analyses
Simulation Center
Alexandria, Virginia

ABSTRACT

The rapid assimilation of leading-edge technologies into the modeling and simulation (M&S) tools of today has provided a revolutionary approach for many of DoD's users, particularly within the acquisition, joint training and joint experimentation communities. However, leveraging technology to assist the warfighter with the "process of applying incremental review, analysis, evaluation, and testing to M&S products to improve credibility" (Defense Modeling and Simulation Office, DMSO, definition of V&V process) has not kept pace. One area of M&S that has undergone the greatest metamorphosis is the synthetic environment. In the short span of 10 years, what was a paper-like digitized terrain map depicting a flat earth at "high noon" and devoid of weather has evolved into a sophisticated synthetic natural environment. Modern simulation synthetic environments continue to grow in size, geodetic rigor, geographic fidelity, and application domain. The richness of these data sets, while markedly improving their utility and widening their applicability, increases the likelihood of error caused by feature interaction and the sheer complexity of construction techniques. Although our capacity to produce large and complex digital environments has seen great improvement, there has not been an attendant growth in the development of automated tools that can help ensure credible results. This paper addresses V&V of federated synthetic environments. The first section exemplifies classes of digital environment conditions that have proven to contribute to anomalous simulation entity behaviors or incorrect analytical results. We describe a utility to automatically locate such conditions as the first step toward managing inter-simulation interactions within a federation. Then, we discuss a methodology to assess interaction validity of the environmental algorithms and data, across a federation. Finally, we describe this process as an asset for event managers, aiding in the design of scenarios that will avoid unfair fight situations.

ABOUT THE AUTHORS

Robert Richbourg is a member of the Research Staff in the Simulation Center at the Institute for Defense Analyses. He is a retired Army officer who in 1987 earned his Ph.D. in computer science. In his last active duty assignment, he was the Director of the Artificial Intelligence Center at the United States Military Academy, West Point. He has been working in the area of simulation environments for 10 years, under sponsorship of DARPA, DMSO, and STRICOM programs. He is currently the project leader for the SEE–IT development effort, the Chair of the SISO product development group responsible for standardizing the SEDRIS Environmental Data Coding Specification, and the technical lead for the DMSO Environment Federation project.

Bob Graebener is a member of the Research Staff in the Simulation Center at the Institute for Defense Analyses. He retired from the military in 1997 after 25 years of combat arms and operations research related assignments. His last active duty assignment was as the Chief of Modeling and Simulation at the USACOM Joint Training, Analysis and Simulation Center (JTASC), where he was the operational manager for the STOW Advanced Concept Technology Demonstration. He is currently a team lead with the Joint Semi-Automated Forces (JSAF) program at IDA. In that role, he has participated in all the major joint experiments sponsored by USJFCOM. He is currently working towards a doctoral degree from George Washington University.

Tim Stone is a Research Programmer/Analyst in the Simulation Center at the Institute for Defense Analyses. His interests include: Computational Geometry, Triangulated Irregular Networks (TINs), Simulation, Algorithm design, Line of Sight (LOS), and Computer Graphics. He obtained his Master's degree in Computer Science from New Mexico State University in 1996. He is a co-developer of the SEE-IT application.

Keith Green is a member of the Research Staff in the Simulation Center at the Institute for Defense Analyses. He holds a master of engineering degree in engineering math and computer science from the University of Louisville and is currently working towards a Ph.D. at George Mason University. He has been working in distributed simulation for the past 12 years.

VERIFICATION AND VALIDATION (V&V) OF FEDERATION SYNTHETIC NATURAL ENVIRONMENTS

Robert F. Richbourg, Robert J. Graebener, Tim Stone, and Keith Green
Institute for Defense Analyses
Simulation Center
Alexandria, Virginia

PROLOGUE

It is time for the modeling and simulation community to develop automated tools that facilitate V&V.[1] Today, simulations such as the Joint Semi-Automated Forces (JSAF), used by United States Joint Forces Command (USJFCOM) J9, Directorate of Joint Experimentation, provide a robust environment, including complex entities and behaviors. Traditional assessment of the entire simulation environment is virtually impossible. Experience has shown that informal V&V techniques, such as employing subject matter experts to study results and parameters, can only address a small fraction of the possible interactions. In addition, advances made in synthetic terrain (or, on a larger scale, the synthetic natural environment) complexity and content preclude any subject matter expert from asserting that environmental accuracy meets the operational requirement.

Although tremendous effort has been expended on developing V&V processes and recommended practices[2] there are few state-of-the-art automated tools to assist in carrying out this critical function. Equally important, the community needs answers to the questions regarding underlying issues that await those ready to apply automated agents in the V&V solution.

An effort undertaken by the authors and J9, USJFCOM, provides some insight. Conducted during the winter and spring of 2001, the initiative targeted a much overlooked area of simulation V&V, synthetic terrain[3] and leveraged a tool developed as part of the Synthetic Environment Data Representation and Interchange Specification (SEDRIS) program sponsored by the Defense Modeling and Simulation Office, DMSO. The intent was to expose the layers underlying synthetic terrain to identify the subcomponents of the synthetic natural environment which potentially could (1) cause an adverse impact on the simulation entities that are normally the subject of V&V inspections and (2) induce bias into the observed outcomes.

The warfighter of today is versed on the impact terrain plays on military operations; tomorrow's warfighter must be cognizant of the impact synthetic terrain has on simulation-supported joint training and experimentation events. We must leverage lessons learned to develop methodologies and automated tools that ensure simulation result relevance, eliminating any doubts that could be attributed to the simulated environments alone.

INTRODUCTION

Composition of multiple simulation systems into a single simulation federation is becoming much more commonplace as current simulation activities strive to include greater breadth, moving away from single purpose, service-centric events. However, the promise of enriched experience by federating heterogeneous simulation systems can be negated when the interoperation of those systems becomes at risk because of different environmental data or uses of that data within the individual systems. As an example, suppose that simulation entity E_A, controlled by simulation system A, is within engagement range of simulation entity E_B, represented natively in simulation system B. Further, suppose that simulation A applies its own line-of-sight[4] algorithm to its native terrain representation to determine the ability of E_A to acquire a target at the location of E_B. Similarly, simulation B will rely on its

---

[1] _Verification_—The process of determining that a model or simulation implementation accurately represents the developer's conceptual description and specification. _Validation_—The process of determining the degree to which a model or simulation is an accurate representation of the real world from the perspective of the intended uses of the model or simulation.

[2] The Defense Modeling and Simulation Office's *Verification, Validation and Accreditation (VV&A) Recommended Practices Guide*, Nov 96, is an excellent example.

[3] Current simulations are moving away from the static, flat-earth backgrounds to robust synthetic environments offering numerous interaction implications.

[4] In the paper, line of sight refers to geometric intervisibility between two point bodies of mass. Computationally, line of sight is assumed to exist unless a portion of the terrain surface, or a surface feature, intersects, in three-dimensional space, the imaginary line segment between two points referred to as the observer and target. Further, this type of line of sight is always reflexive; if the target is visible to the observer, then the observer is also visible to the target.

native resources (algorithms and data) to reach its own conclusion on the existence of line-of-sight between E_A and E_B. If the two simulations use different algorithms or different data, it is very possible that they will reach different conclusions on entity-to-entity line-of-sight. In such a case, E_A might acquire and engage E_B, while E_B is not aware of E_A, even after engagement.

This is a simple example of problems that might arise in federated simulation systems. Whenever different federates reach different views of the same situation, some federate may allow unfair advantage to the users it services. This is the "unfair fight" problem that can arise during the interoperation of heterogeneous simulation systems. Although several simulation system characteristics can contribute to existence of unfair fights, the preceding example illustrates two common causes: (1) system representations of or (2) uses of digital environmental data.

The digital environment representation, often called the Synthetic Natural Environment (SNE), can include data from all environmental domains, describing land, sea, air, or space. The SNE unfair fight problem is not new, and some techniques have been developed to overcome the associated problems. Without proof, we simply assert that two different environment representations cannot be made functionally equivalent without becoming identical. An alternative focuses on managing the intrinsic differences themselves. Two popular management strategies are both intended to preclude situations where two or more simulation systems could develop inconsistent world views. The first separates different simulations in time or space (or both). In this case, different simulations may all operate within the same federation of simulations, but they do not directly interact with each other during any specific simulation event. Using the previous example, E_A would never have an opportunity to interact with E_B. The second management method does allow direct interaction between heterogeneous systems, but requires that one of the interacting systems assume a dominant role by making all decisions about each interaction. Using the above entity-to-entity engagement example, either simulation A or B would determine the existence of line-of-sight and provide the other simulation with its result, thus assuming the role of "line-of-sight server" and ensuring that both simulations reach the same conclusion.

Both of these approaches to resolving the unfair fight problem have the potential disadvantage of diminishing the value of federating systems at all. The former approach prohibits interaction of systems, so the overall simulation could more appropriately be described as a collection of separate systems, since no direct interaction is allowed, used, or required. The second approach is more acceptable, but clearly has shortcomings. Subordinate systems may well be stripped of their essential character when their decision-making authority is removed. Further, dominant systems might incur greatly increased computational load and communicating their decisions would increase network load, adding latency into the federation.

This paper describes alternative methods that neither preclude direct interaction between federates nor force any federate to assume an "information server" role. The approach requires pre-event identification of locations and types of interactions that have a very low probability of introducing unfair fights. The method requires analysis of each SNE representation individually, a collective analysis of the functional uses of those representations, and a comparison of the different algorithms that each federate will apply to reach its own conclusions. These analyses result in information that either permits design of events featuring system interactions in ways and locations that promote fair fights or advises the simulation manager that suitable conditions for such interaction do not exist.

DATA ACCESS AND SEMANTICS

Very few simulation systems share identical (internal) representations of their environment data, although some notable exceptions exist. As an example, both the Joint Warfare Simulation (JWARS) and Joint Semi-Automated Forces (JSAF) rely on the Compact Terrain Database (CTDB) format for representing their SNE data.[5] Examples of systems that use different representations are far more numerous. The Joint Simulation System (JSIMS) databases include buildings that are described as point features with length, width, height, and angle of orientation attributes. Buildings are represented in S1000 format data by point instances that specify the rotation, scaling, and translation to be applied to specific three-dimensional models. Some systems, such as the Close Combat Tactical Trainer (CCTT), rely on multiple polygonal descriptions of the terrain surface, each appropriate for a specific level of detail (LOD) view of that surface. In comparison, other systems, exemplified by the Simulation of the Locations and Attack of Mobile Enemy Missiles (SLAMEM), rely on direct representations of National Imagery and Mapping Agency (NIMA) Digital Terrain Elevation Data (DTED) to describe a single terrain surface. Locations within NIMA products are typically

[5] The STRICOM-sponsored OneSAF program also seems to be converging on the CTDB format for its future SNE developments.

represented in geodetic coordinates, while JSAF simulations use several representations, including augmented Universal Transverse Mercator (UTM) or the Global Coordinate System (GCS).

Any system or methodology designed to compare and correlate like data elements across a wide space of diverse, native formats, such as the few noted above, has two significant problems. First, multiple access mechanisms are required. A software method to reach into an S1000 database will not be useful when a CCTT database must be read. Second, different internal representations of data lead to potentially different interpretations of that data when attempting to correlate like data elements. As an example, a strategy that interprets appropriate line segments as roads when reading a CCTT database will not succeed when applied to a WARSIM database, especially if bridges are involved. Solving these two problems is a preliminary requirement to federate correlation. Clearly, engineering such a solution from scratch would be very costly in terms of time and resources.

Fortunately, a suitable solution has been in development since 1995. The SEDRIS program was initiated under Defense Advanced Research Products Agency (DARPA) sponsorship and continues as a DMSO-led effort to reduce government expenditures on environment database construction by allowing reuse of such data. SEDRIS is intended to enable the unambiguous representation and efficient interchange of all forms of digital environmental data. Progress has been significant, and SEDRIS is well on the way to recognition as the international standard (through the International Standards Organization, ISO) in this area. The SEDRIS Data Reference Model (DRM), Spatial Reference Model (SRM), and Environmental Data Coding Specification (EDCS) will all become individual ISO standards that respectively permit the unambiguous representation, location, and identification of SNE data. Efficient access is provided by application program interfaces into SEDRIS Transmittal Format (STF) files that encode data according to the SEDRIS DRM. Without further discussion, we assert that SEDRIS is the most appropriate solution to access federate data, and SEDRIS provides this capability as an existing infrastructure technology, without requiring new development effort. SEDRIS management has often noted that the program is neither designed nor implemented as a solution to the interoperability problem. However, SEDRIS is a necessary precursor to the development of such solutions. Our use of and reliance on SEDRIS follows this line of reasoning exactly.

METHODOLOGY OVERVIEW

The introduction describes unfair fight issues that might arise from differences in federate line-of-sight calculations and notes that several other factors could lead to similar situations. In fact, this situation has been elegantly described by Birkel (1998) as he developed the SNE Conceptual Reference Model (CRM), illustrated in Figure 1. This model provides a conceptual view of the transformational relationships between SNE data and simulation outcomes. Note that each simulation within a federation may have some or all of the illustrated components and that each of these components has an opportunity to influence the simulation outcome. SNE-related issues drive our view of avoiding unfair fight problems among interoperating systems, so our use of CRM focuses on the Data and Effects portions of Environmental Ground Truth. The methodology we propose herein attempts to identify, isolate, and examine independent aspects of SNE-related interoperability. As an initial step toward correlation of databases between federates, we require that each federate's native environmental database be correct within its own context of use. More specifically, successful interoperation between systems requires that no simulation entity or aggregate exhibit anomalous behavior, driven by errors in their native database that would also occur if the simulation was used in isolation, apart from any federation.

Our methodology relies on 3 axes to characterize correlation: feature content and placement, line-of-sight, and mobility. Of the three, feature content and placement is the easiest to understand. Whenever two or more federates include different features or place identical features differently, poor correlation of simulation outcome is possible. Thus, our approach requires the direct comparison of the feature data within each pair of federates, including type, size, extent, orientation, and location. Resulting information includes a description of agreement by feature and by federate pair. This comparative data allows simulation managers to design appropriate (feature-related) federate interactions, initiate changes to federate databases to produce agreement, or simply avoid inappropriate interactions.

The other axes for correlation, treatment of line of sight and mobility, have somewhat more complicated implications, and the associated methodology is fully described in the following sections. We illustrate the problems and solution using only line-of-sight comparisons, but note that mobility is also a fundamental concern. In general, our approach requires examination of related data and algorithms and their

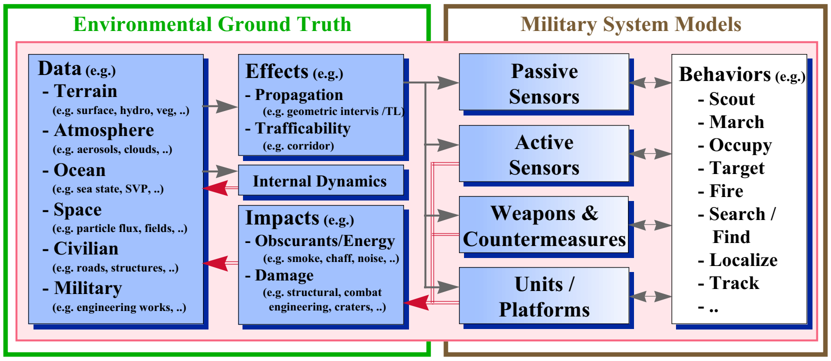

Figure 1. SNE conceptual reference model (Birkel, 1998)

interactions. A constant set of problems is solved according to the data and algorithms native to each federate, resulting in comparative data for use by simulation managers, who are then able to design federate interactions that have the highest likelihood of resulting in fair fights

DATABASE CORRECTNESS

Internal database content and constructions can result in anomalous simulation behaviors, even when the simulation is used independently of any federation of systems. Figure 2 shows a ModStealth screen capture of ModSAF entity behavior as the force attempted to move across a terrain surface that was constructed using an unexpected topology. Typical ModSAF databases feature polygonal surface constructions where each polygon is expected to share vertices with its immediate neighbor polygons. Given this construction, each surface polygon can only have one neighbor polygon per edge. In some cases, a vertex of one polygon might lie on the edge of its neighbor polygon, resulting in the occurrence of a "T-vertex." This construction results in multiple neighbor polygons across a single edge (see Figure 3). This type of construction resulted in the entity behavior shown in Figure 2. As the ModSAF vehicles began to fly or burrow, they also ceased responding to any commands. They may as well have been casualties of direct fire. If similar behaviors occurred during a federation event, opponents would have a clear unfair advantage, effectively destroying all of these simulation entities without any engagement.

In practice, there might be many types of constructions and conditions within a database that can lead to undesirable simulation results. For example, vertical tears or holes in the terrain surface allow simulation entities to position themselves "under" the terrain, affording them a unique and powerful concealment option. Disconnected road networks can confuse simulation entities. Again, ModSAF entities have been

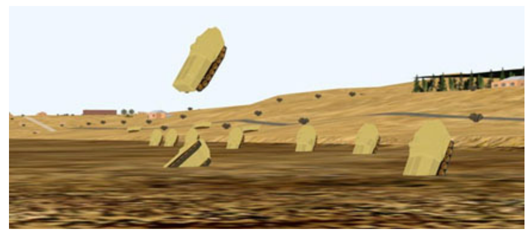

Figure 2. Anomalous ModSAF behavior

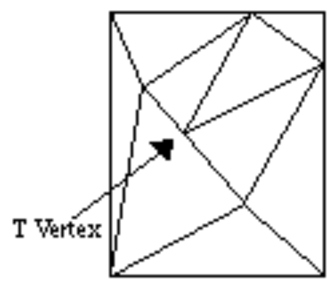

Figure 3. T-vertex

observed stopping when commanded to road march along such networks. Other entities have received major damage after collisions with invisible model bounding volumes when these volumes were incorrectly specified. Note that these conditions only become pathological within specific use contexts. A CCTT entity, for example, is not sensitive to T-vertex constructions.

The SNE development community has long been aware that unintended database constructs can contribute to (or directly cause) anomalous simulation outcomes though their influence on entity behavior. However, identification of such constructs is not a trivial task. The topological construction (Figure 2) in question is not visible to the human observer. Moreover, context of use plays a major role. Thus, one attempt to locate such conditions has relied on exhaustively sending simulation entities back and forth across the database, waiting for something unexpected to occur. This approach has long since outlived its utility, given the continuing trend of producing and using ever larger, more complex simulation-environment databases.

As an alternative, the SEDRIS program provides an automated tool that can be used to locate specific geometric constructs within individual environmental databases. This utility, the Synthetic Environment Evaluation—Inspection Tool (SEE–IT), has been available to SEDRIS associate programs since 1998. It is currently an integral component in the environmental data construction process for WARSIM, debuted as a potential V&V tool for the recently completed Unified Vision joint experiment, and has been adopted by multiple database construction enterprises. Like other SEDRIS software, SEE–IT is available at no cost (for more information visit *http://www.sedris.org*).

A complete description of SEE–IT is available elsewhere (see Richbourg and Stone, 2001). Here, we note that it constitutes a viable method to help achieve a necessary precondition for federate interoperability: each federate must have a correct database. SEE–IT does not address the complete problem, however. Although it can be used to locate and identify specified conditions, it does not, as yet, fully address the correction of those conditions. Still, knowledge of database areas that might produce unwanted simulation outcomes is valuable information for an event manager. If associated conditions cannot be corrected, some parts of the databases can be avoided, using only known correct parts of the SNE to conduct the simulation event.

SEE–IT does include an application program interface that other software systems can use to access identified error conditions. SEE–IT can also automatically correct a small number of the conditions that it locates. The software will automatically repair T-vertices without changing the fidelity of a database. However, almost all unintended constructions require some human input to determine the most appropriate correction strategy (see Richbourg and Stone, 1998). As an example, if a river lies on the side slope of a valley, the situation could be corrected by moving either the river into the valley or by moving the valley under the river. Thus, it seems that editing will be required in many cases

DATABASE FUNCTIONAL CORRELATION

This component of characterizing federate interoperability is easily understood at a conceptual level; however, the attendant details can become somewhat complex. To simplify, we illustrate using only line-of-sight functions, even though we assume treatment of mobility to be equally important. Accordingly, the exact problem to be solved is as described in the first paragraph of the introduction: what can be done to ensure that two federates will reach the same conclusion, given identical instances of a line-of-sight problem?

There are several contributing factors to be considered. Separate federates might well have different native environmental databases and different line-of-sight algorithms or implementations. Note that uncorrelated results can occur when different line-of-sight algorithms are applied to the same database, the same line-of-sight algorithm is applied to different databases, or when different line-of-sight algorithms are applied to different databases. For simplicity of illustration, we discuss the first two cases below, but note that federation interoperability is most likely to depend on the third case. However, the following discussions are easily extended to address that case.

Same Algorithm, Different Databases

This situation would occur whenever two similar systems have selected the same line-of-sight algorithm, but based their simulation on different environmental databases. This clearly might lead to interoperability difficulty. As an example, consider the 3,100 m terrain cross-sections illustrated in Figure 4. The lower cross-section (presented as the thicker line) is based on elevations derived from NIMA DTED Level 1 source (approximately 84 m horizontal distance between elevation points in this area of the world). The other (top) cross-section represents a surface derived from NIMA DTED Level 2 (approximately 27 m horizontal distance between elevation points). These NIMA files describe exactly the same area, but provide radically different elevation information (elevation in meters is shown on the left side scale). Figures 5 and 6 provide shaded relief views of the region where the cross-section view was sampled; the black line identifies the cross-section itself.

Suppose that simulation federate A has terrain data derived from DTED 1 and federate B has terrain data derived from DTED 2 source data. Further, suppose federate A has an entity located at the position marked by the left vertical line in Figure 4 (at about 1546 on the bottom scale) and that federate B has an entity located at the location of the right side vertical line (between 2320 and 3093 on the bottom scale). Again, assume that both federates rely on the same line-of-sight algorithm. Federate A will calculate a blocked line of site between these locations (based on DTED 1), while federate B will declare line of site exists (derived from DTED 2) and allow an engagement to occur. This is clearly

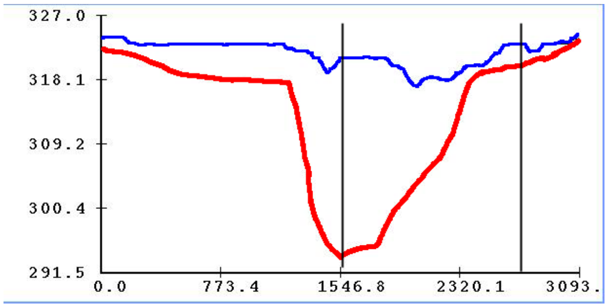

Figure 4. DTED 1 and DTED 2 terrain elevation cross-sections

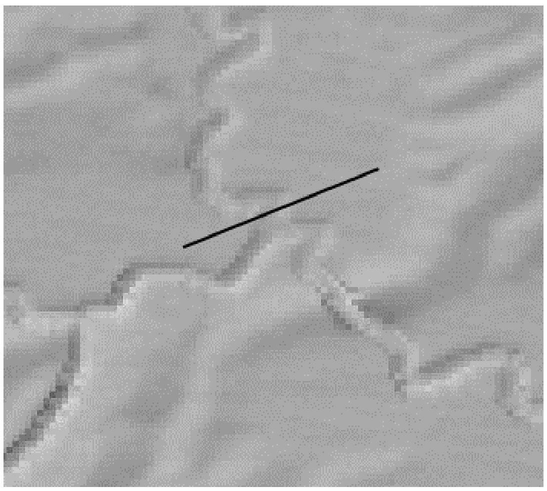
Figure 5. DTED1 shaded relief

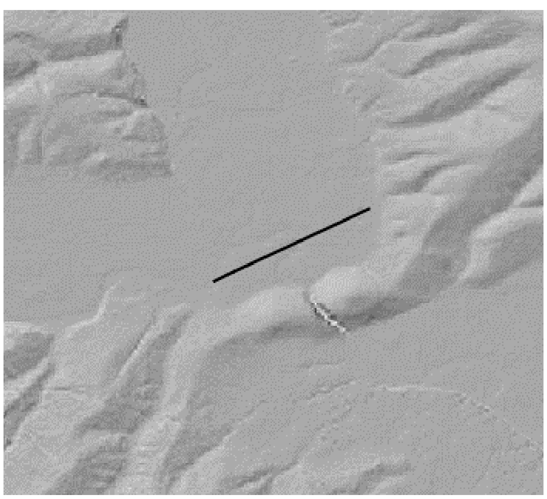
Figure 6. DTED 2 shaded relief

an unfair fight situation, despite both federates' use of authoritative data and the same line-of-sight algorithm.

Different Algorithms, Same Database

This situation might occur when federated simulations that share a database format (such as JWARS and JSAF) have chosen different line-of-sight algorithms, or permit simulation users to select or modify the algorithm to be used (as is the case for JSIMS). There are two classes of line-of-sight algorithms that can lead to inconsistent results. One class logically modifies the terrain during the calculation. These algorithms are typified by a subset of those that make their calculations based on a regular grid of elevations. In one typical implementation, the calculations posit the existence of a three-dimensional line segment between the observer and target and then inspect for intersections with all of the three-dimensional line segments that run exactly through the elevation points and parallel to the x or y axis.

Figure 7 provides an illustration of this simple case. Here, a thin-line, regular grid represents the line segments that could be derived as passing through elevation data points and parallel to either the x or y axis. The observer-to-target line is illustrated as a thicker diagonal line, and intersection points with the regular grid are shown as heavy black circles. Here, the line-of-sight between observer and target would be determined based on the inspections made at eight intersection points. Elevations along the grid lines can be calculated using several methods. These typically include reference to the elevation value at the closest grid point, linear interpolation between the grid points at the segment intersection, and bi-linear interpolation using the nearest four grid points. Because each of these methods might determine a different elevation value, line-of-sight results can also vary based on this choice. However, other more dramatic differences among algorithms in this class arise due different grid-sampling strategies.

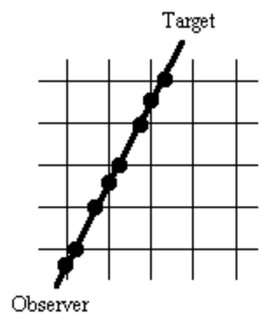

Figure 7. Simple grid-based line-of- sight intersections

More accurate versions of this same basic algorithm include an additional step before determination of the line-of-sight result. Here, diagonal line segments between elevation grid points are also considered as providing opportunities to block the line of sight. Obviously, there are two diagonal lines that meet this criterion, and either or both could be chosen for reference during this part of the calculation. This choice has a fundamental impact. Figure 8 shows the same line-of-sight problem as used in Figure 7, but adds diagonal grid line segments. For clarity, Figure 8 does not repeat the test points shown earlier, indicating only the additional points for testing line of sight based on diagonal grid lines. Note that one case adds only three additional points while the other requires testing an additional eight points.[6] Clearly, there is ample opportunity for these different implementations to reach different conclusions, even when using exactly the same elevation data. These differences are not due entirely to the number of points to be tested.

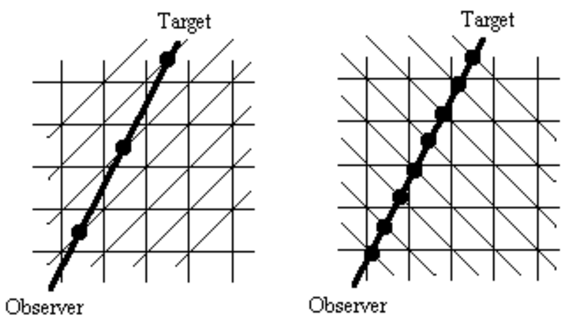

Figure 8. Additional test points derived from grid diagonal line segments

Potential differences arise in part because the choice of the reference diagonal is really a basic interpretation of the terrain itself. Consider four adjacent elevation grid points that collectively form a square (in Cartesian space) and assume that the southeast and northwest corners are at elevation 10 while the other points are at elevation 20. Referencing the southeast to northwest diagonal (lower elevation, Figure 8, right) transforms these points into a "local valley." Referencing the opposite diagonal (higher elevation, Figure 8, left) transforms these points into a "local ridge." These are opposite interpretations of exactly the same data!

These differences arise even when using the very similar algorithms we posit above. Comparing widely divergent algorithms usually results in differences based on fundamental computation rather than on interpretation of the terrain. Recently, the JSIMS program considered the merits of using a computationally inexpensive line-of-sight estimation method as opposed to an exact algorithm. Both algorithms were to be applied to the polygon set describing JSIMS terrain surface and features on that surface. This estimation technique samples terrain and features at uniform, prescribed intervals along the line of sight. The exact method inspects the edge of each polygon (and feature) for potential intersections that could block line of sight. This algorithm was assumed to be less time-consuming because it based decisions on fewer line-of-sight intersection inspections. The Institute for Defense Analyses assisted in strategy selection by performing experiments implementing the two methods and comparing their performance using the JSIMS SNE. Of interest here are the accuracy comparison results, including those pertaining to selection of a suitable sample interval for the estimation method.

The comparison was based on solving approximately 80,000 line-of-sight problems that intersected representative portions of the JSIMS SNE. Observer and target heights above terrain were also varied to emulate expected engagements in the simulation (e.g., ranging from ground-to-ground tank engagements to air-to-air engagements). These tests were completed for the exact method and 20 instances of the estimation method, each of which used a different sample interval. The study concluded that the importance of both choice of algorithm and terrain character decreased as either the target or observer (or both) increased height above surface. Also, the estimation method accuracy decreased as sample interval increased. Finally, the study documented and quantified an unusual characteristic of the estimation method: this approach does not always preserve line-of-sight reflexivity, so that even within a single simulation (using only one algorithm and one SNE), unfair fights could result when the target is visible to the observer, but not the converse. Sample points based on a sample interval referenced to the target location can be different from the points chosen when the observer location is the reference point.

Figure 9 presents typical results, based on potential tank-to-tank engagements over a 3 km range. The horizontal axis denotes the various sample intervals (ranging from 50 to 1,000 m) while the vertical axis denotes percent agreement. The points labeled "Reflexive Agreement" (open diamonds) refer to the estimation method's consistency with itself. This data includes several sample intervals that return to near-complete agreement, even though the overall trend seems to monotonically decrease with increased sample interval. In fact, the sample intervals that provide near agreement, regardless of interval size, are those that exactly divide (no remainder) the line-of-sight distance (3 km here). In these cases, initial reference point is immaterial.

[6] Adding more test points to a line-of-sight problem cannot restore a blocked line of sight, but increased testing might result in a block where none existed previously. Thus, testing a greater number of points during a line-of-sight calculation generally produces more blocked lines of sight.

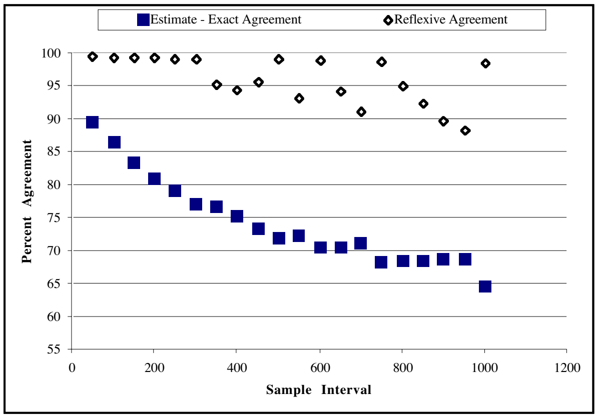

Figure 9. Estimate and exact method comparisons, 3 km line-of-sight problem for armor

The other Figure 9 point set (filled squares) relates the estimation method accuracy to that of the exact method. The example problem represents a low case of algorithm agreement, ranging from 65 to 89%. Agreement between algorithms improved with target height above surface. A low-flying helicopter to tank engagement ranged from 67 to 91% agreement, and a tank to fixed wing aircraft engagement ranged from 76 to 93% agreement.

MANAGING DIFFERENCES

Recall our assertion that attempts to establish federate correlation by modifying either SNE databases or algorithms are essentially futile (unless one becomes identical with the other). Instead, we prefer a strategy that manages the SNE-related differences between federates. Such management depends on information that associates a probability of unfair fight occurrences with specific locations in the database, given specific types of federate interactions. As an example, suppose the existence of federate system A where SNE data is exactly DTED level 1 and that federate B relies on a polygon surface representation derived from DTED level 2. Federate A uses a basic line-of-sight algorithm appropriate for grid elevation data; federate B relies on an exact polygon algorithm. These differences will be completely immaterial if both SNE describe a flat desert area. However, if the surfaces appear more similar to those illustrated in Figures 5 and 6, unfair fights between these two federates are certain to occur. One method to quantify the functional differences would be to apply both algorithms to their native data for each possible line-of-sight problem and then compare the results.

Although such an approach provides clearly desirable results, it is computationally intractable, given SNE data of typical extent. However, we can approximate the desired characterization. First, the problem can be decomposed into instances that reflect likely engagements (i.e., reflecting specific range and height above surface for target and observer). Second, we can modify and then apply a line-of-sight estimation technique that has been successfully used in related problem domains (see Ray, 1994, and Richbourg & Olson, 1996). In fact, we are now developing a SEDRIS-based system along exactly these design guidelines. This system includes a collection of line-of-sight (estimation) implementations, an ability to ingest and reason about SEDRIS-format SNE databases, and a capability to apply selected algorithms (user-tailored for likely engagements) to individual databases.

In operation, one can select a line-of-sight estimation algorithm (from the system catalog) that simulates the operation of the basic grid algorithm (of federate A) described above. This algorithm could be applied to a SEDRIS form of the federate A database. Similarly, the operation of the exact polygon algorithm on the federate B database could be estimated. The system can also compare the results from the two estimates, over a space of varied line-of-sight problems, using ranges and heights above surface appropriate for tank-to-tank engagements, air-to-air engagements, or any combination in between. The system provides two useful products for each specified engagement type. The first is a graphic, unique for each database, depicting relative line-of-sight potential for each shared location in the two databases. The second is a graphic overlay that describes probability of fair fights for each shared location in the two SNE databases.

Figures 10 and 11 provide shaded relief representations of a single region in Kosovo, one derived directly from DTED level 1 and the other from DTED level 2. These images represent exactly the same 30 by 30 km region. There are some clear differences in these two representations, aside from the much better portrayal of detail in the higher resolution DTED 2 data. The DTED 1 shows the presence of "cornrows" in the flat areas (caused by anomalies in the collection process) as well as expanded drainage features that appear to have been edited by hand (also see Figures 5 and 6).

Figure 12 illustrates the first system product, a visibility map that has been derived from an armor engagement analysis of the DTED level 1 area. Here, lightly colored areas depict regions that offer relatively superior lines of sight and fields of fire out to a range of 3,000 m. The darker the area, the worse this potential, so that very dark areas depict concealed locations for armor (per opposing armor observation). These depictions are intuitively appealing. The "flat" areas seem to be the

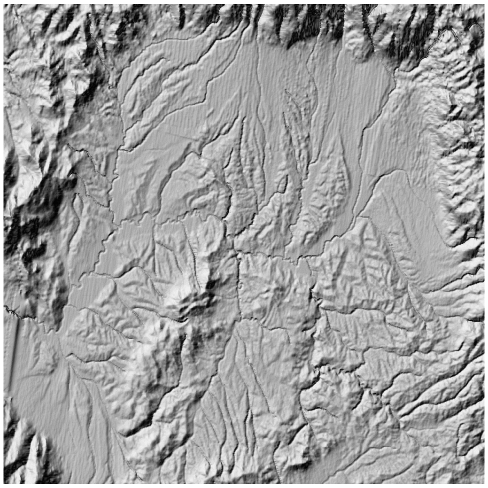
Figure 10. DTED level 1, Kosovo region

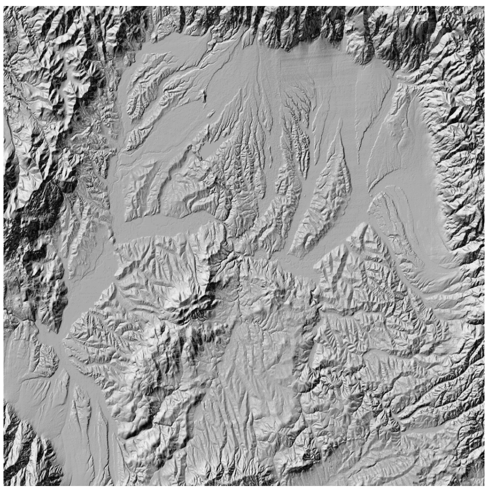
Figure 11. DTED level 2, Kosovo region

most likely high-engagement areas, and they are the lightest-colored areas in Figure 12. Such information is useful in that it suggests areas where simulation managers should concentrate their forces to either allow or deny engagements. The system produces one of these visibility maps for each database, by engagement type (i.e., the air-to-armor visibility map, derived from the same database, is different).

As a byproduct of calculating the visibility maps, the software also constructs graphic overlays that characterize the probability for fair fights, by engagement type, for each pair of databases. For each individual line-of-sight estimate, the system compares the distance over which system-specific algorithms applied to system-specific databases agree on the result. This distance is directly translated into a probability of both federates calculating the same line-of-sight result (and thus producing fair fights). Figure 13 illustrates this product, comparing the potential of the DTED 1 and

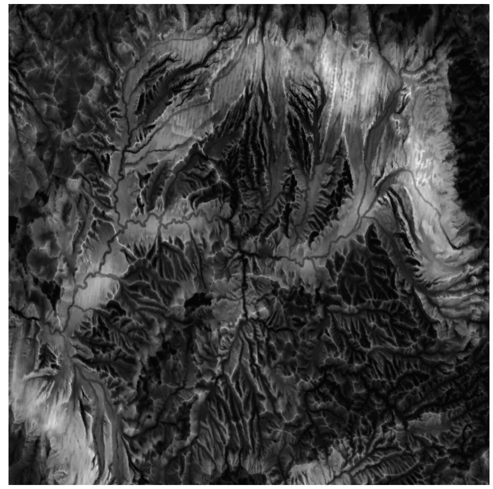
Figure 12. DTED level 1 armor problem visibility map

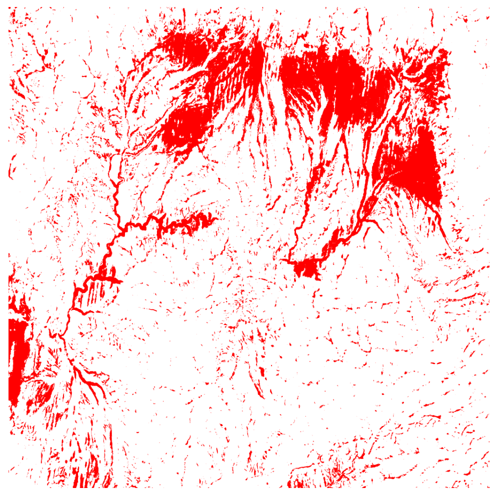
Figure 13. Areas likely to support fair fights (in white)

DTED 2 databases, each using different grid-based, line-of-sight algorithms, for potential armor-to-armor engagements. Here, areas that offer less than 90% potential agreement are shaded, and areas promising 90% or greater potential for agreement are shown in white. (Any percentile can be selected for display.) Note that the areas most favorable for engagements (near white in Figure 12) generally offer poor correlation potential (shaded in Figure 13). Even so, simulation managers could use the available information to find promising areas for federate interaction or could eliminate interaction types (e.g., armor engagements in Figure 13) if insufficient favorable engagement areas exist. Similar overlays are available for other engagement types, allowing managers to select the most appropriate type of federate interactions or conclude that no reliable interaction is possible, based on the contemplated engagement types.

CONCLUSION

Ensuring fair fights in federation events when systems use different SNE data and algorithms is far from trivial. The first significant problem is that of sharing and understanding federate SNE data. The SEDRIS program provides an invaluable capability here, allowing the unambiguous exchange of all federation SNE data with an external analytical system that can provide estimates of likely federation performance. Currently, this analysis system is in early development. Eventually, it will be available as no-cost SEDRIS software to aid in the conduct of large federation events. The system will see its initial critical use with USJFCOM's Millennium Challenge event in 2002.

Every component of the SNE CRM, including data and the algorithms applied to that data, has some influence on the SNE fair fight problem. Our approach requires initial efforts to verify that each federate has an error-free SNE database. Then, the SNE data must be examined within the context of its intended use and compared across federates. Doing so permits an assessment and validation of appropriate types of federate engagement interactions, including identification of locations where those engagements are likely to result in fair fights, if such locations exist. This approach does not make any attempt to bring either SNE data or algorithms into a correlated state. Instead, it is intended to arm event managers with the information that will allow them to design a successful federation simulation without changing any federate data or implementations.

Significant work remains to be done. The methodology we propose addresses only a portion of the SNE data currently in use. Important components, typified by atmospheric, weather, and hydrologic data, have yet to be incorporated. Related issues will be included in future efforts, resulting in a utility that should help ensure credible federation events using the full spectrum of SNE data.

ACKNOWLEDGEMENT

The authors thank Mr. George Lukes, IDA, for his early contributions during concept development.

REFERENCES

Birkel, P. (1998) SNE conceptual reference model. *Proceedings, Simulation Interoperability Workshop, Fall 1998*. paper number 98F-SIW-018. http://www.sisostds.org/confandwork.cfm (2001, May 29).

Ray, C. (1994). *Representing visibility for sighting problems*. Ph.D. thesis, Rensselaer Polytechnic Institute.

Richbourg, R. and Olson, W. (1996). A hybrid expert system that combines technologies to address the problem of military terrain analysis. *International Journal of Expert Systems with Applications*. Vol. 11, No. 2. 207-225.

Richbourg, R. and Stone, T. (1998). Automating error detection and correction in synthetic environments. *Proceedings, Simulation Interoperability Workshop, Fall 1998*. Paper number 98F-SIW-087. http://www.sisostds.org/confandwork.cfm (2001, May 29).

———. (2001). *Synthetic environment evaluation—inspection tool (SEE–IT) User guide*. Contact http://www.sedris.org (2001, May 29).