A Unit-level Combat Resolution Algorithm Based on Entity-level Data

Mikel D. Petty
University of Alabama in Huntsville
301 Sparkman Drive, Huntsville AL 35899
pettym@uah.edu

James Panagos
Gnosys Systems
198 Broadway, Providence RI 02903
jpanagos@gnosyssystems.com

ABSTRACT

Unit-level combat models provide computational efficiency, with the result that they can simulate large scenarios in terms of geographic scope and size of military forces involved and are often able to execute much faster than real-time. However, existing unit-level combat models (such as Lanchester equations) don't exploit the detailed performance data and high-fidelity models that are available at the entity level. In entity-level combat models combat phenomenology, such as moving, sensing, and shooting, is represented at entity level, which is both more intuitively acceptable to users and more directly supportable by available test and operational data on entity performance than the abstract equations of a unit-level model. However, current pure entity-level combat models tend to produce unrealistically high attrition.

Under DARPA sponsorship, we have developed an alternative unit-level combat resolution algorithm. Within it the effects of moving, sensing, and shooting on unit-level combat outcome are based on entity-level performance information, directly supportable by test and operational data. Despite entity-level basis, the algorithm is sufficiently abstract to allow responsive execution in the context of a unit-level simulation. Entity-level performance is represented by a set of probability functions that eliminate as much entity level detail as possible while retaining the important effects of entity-level performance on combat outcome. These probability functions, all of which are based on entity-level data and models, include intervisibility, detection, kill, and location. The functions operate within equations that consider potential interactions between entity types and likely locations of entities of different types.

This paper will explain the combat resolution algorithm, its basic equations, the probability functions, and how the latter are based on entity-level data. It will also report work to test the new algorithm by comparing its results with the outcome of a historical battle.

ABOUT THE AUTHORS

Mikel D. Petty is Director of the University of Alabama in Huntsville's Center for Modeling, Simulation, and Analysis. He received a Ph.D. in Computer Science from the University of Central Florida in 1997. Dr. Petty has worked in modeling and simulation research and development since 1990. He has published over 135 research papers and has been awarded over $12 million in research funding. He served on a National Research Council committee on modeling and simulation, is a Certified Modeling and Simulation Professional, and is an editor of the journals *SIMULATION* and *Journal of Defense Modeling and Simulation*.

James Panagos is President and Founder of GNOSYS, Inc. He received a M.S. in Computer Science from the Massachusetts Institute of Technology in 1985. He has worked on a range of projects in semi-automated forces, multi-resolution simulation, and virtual training simulations.

A Unit-level Combat Resolution Algorithm Based on Entity-level Data

Mikel D. Petty
University of Alabama in Huntsville
301 Sparkman Drive, Huntsville AL 35899
pettym@uah.edu

James Panagos
Gnosys Systems
198 Broadway, Providence RI 02903
jpanagos@gnosyssystems.com

INTRODUCTION

Unit-level (or aggregate) combat simulations, such as WARSIM, have certain desirable characteristics. Their aggregate representations provide computational efficiency, with the result that they can simulate large scenarios in terms of geographic scope and size of military forces involved and are often able to execute much faster than real-time. However, they don't exploit the detailed performance data and high-fidelity models that are available at the entity level. Entity-level (or disaggregate) combat simulations, such as OneSAF, have a different set of advantages. The level of resolution of their models of combat phenomenology, such as moving, sensing, and shooting, is typically at the entity-level, which is both more intuitively acceptable to users and more directly supportable by available test and operational data on entity performance than the abstract equations of a unit-level simulation. However, current pure entity-level combat models tend to produce higher than expected attrition. Possible phenomena present in actual combat and accounted for in unit-levels but not entity-level combat models that could explain this include target duplication, shooter non-participation, suppression effects, self-preservation, and suboptimal use of weapons and targeting systems.

There has been a longstanding desire to combine these two classes of simulation so as to realize the best features of both in a single simulation system. One approach to doing so, the linking of unit-level and entity-level simulations into multi-resolution systems, has been implemented several times since 1992 in a variety of combinations of specific unit-level and entity-level simulations (Franceschini 1995). Some common elements have emerged in these systems; for example, the operations of disaggregation, wherein a unit represented in and controlled by the unit-level simulation is instantiated as a set of entities represented in and controlled by the entity-level simulation, and aggregation, where the reverse occurs. Despite the common elements, the multi-resolution simulations implemented to date must be considered to varying degree to be point solutions, with algorithms and models specific to the simulations being linked. Moreover, these linkages have several disadvantages inherent in the way they are constructed (more on this later). Finally, there are fundamental issues of validity inherent in such linkages, perhaps most notably that of combat results correlation error, where the outcome of a combat differs in a significant way depending on whether that combat is resolved at the unit level or at the entity level (Franceschini 1999).

Attempting to have unit-level and entity-level representations interact directly with each other is even more problematic. Unlike the familiar and well-understood mathematical models of combat at the unit level and the natural resolution and data supported models of combat at the entity level, there is no theoretical or experiential basis for direct inter-level interactions. Indeed, one significant design study examined four types inter-level interactions (direct fire, indirect fire, command and control, and communications and emissions); that study found that the best available mechanism for implementing such interactions was a form of disaggregation (specifically, pseudo-disaggregation) (Petty 1998).

In this research we intend to develop a means of combining unit-level and entity-level combat simulations that combines the best features of both and avoids the problems and overhead of multi-resolution simulations and inter-level interactions. The essential idea is to develop new alternative aggregate-level algorithms for key combat phenomenology (moving, sensing, and shooting) that are based on entity-level models, with their associated natural entity level of resolution and direct supportability by data, but have been abstracted to allow their responsive execution in the context of a unit-level simulation. These new algorithms will eliminate the need for aggregation, disaggregation, and entity control handoff, at least for purposes of resolving unit-level combat.

BACKGROUND

This section provides brief background information on unit-level and entity-level combat models.

Unit-level models

The results of engagements between aggregate units in unit-level simulations are produced using aggregate attrition models. Distinct activities such as target acquisition and lethality assessment are combined into the attrition calculations. Individual entities are not typically represented in these units, so details of entity-entity engagements are not modeled; instead the attrition process models consider average results. The contributions of the individual entities to the combat's outcome are averaged over the entire unit (for homogeneous models) or over weapon system classes within the unit (for heterogeneous models).

Aggregate attrition models often use Lanchester equations in various forms (Taylor 1980a) (Taylor 1980b) (Taylor 1981) (Fowler 1996a). Lanchester equations are differential equations describing the rate of change of Blue and Red force strengths X and Y as a function of time, with the function depending only on X and Y. One partly generalized version of the Lanchester equations has the following form (Davis 1995):

$$\frac{dX}{dt} = -K_y X^r Y^S \text{ and } \frac{dY}{dt} = -K_x Y^t X^u$$

where K_x and K_y are the attrition rate coefficients of the Blue and Red force, respectively; and r, s, t, and u are free, time-independent parameters that can be used to "tune" or customize the equations for particular situations (also see (Dare 1971)).

The equations may be extended in various ways, e.g., to include constant reinforcement-rate terms, as well as other effects (Helmbold 1965). There are two special cases of the generalized form of the Lanchester equations; the "square law" corresponds to $s = u = 1$ and $r = t = 0$; the "linear law" corresponds *to* $r = s = t = u = 1$.

$$dX/dt = -K_y Y \text{ and } dY/dt = -K_x X$$

(Square Law)

$$dX/dt = -K_y XY \text{ and } dY/dt = -K_x XY$$

(Linear Law)

The square law is usually taken to apply to "aimed fire" (e.g., tank versus tank) and the linear law to apply to "unaimed fire" (e.g., artillery barraging an area without precise knowledge of target locations). The key feature of the square law is that it describes concentration of fire. For a discussion of non-homogeneous aggregation models, see (Taylor 1980b) and (Fowler 1996c).

The attrition rate coefficients K_x and K_y depend on factors such as the time to acquire a target, the time of flight of the projectile, the single-shot probability of a kill, terrain, weather, and others. Methods for determining attrition rate coefficients include derivation form historical battle data (Dupuy 1995) (Peterson 1953), Bonder-Ferrell theory (Bonder 1967) (Bonder 1970), and Markov Dependent Fire models. Target acquisition requires algorithms for search, screening, and detection (Fowler 1996b) (Koopman 1999) (Washburn 2002). Factors include probability of detection, line-of-sight (or probability of line-of-sight), and area of search. In aggregated models, these factors are usually averaged across weapons systems or combatants. Other factors that influence engagement outcomes may be more difficult to represent in the equations; these include defensive variables such as armor and anti-weapons systems, and proactive behavior such as maneuver and use of terrain cover.

Other aggregate attrition models include the Quantified Judgment Model (Dupuy 1985) (Dupuy 1998), Fire Power Scores and Force Ratios (Anderson 1974), and the ATLAS ground attrition model (Kerlin 1969).

Entity-level models

Entity-level simulations model the combat phenomenology in question (moving, sensing, and shooting) at the level of individual entities. The entity-level models consider the performance characteristics of the specific entity, the effects of terrain (and less often, other environmental factors) on its actions, and the entity's location with respect to other entities. The important point is that the entity-level models are often based on performance data for the specific entity type; that data may be gathered from testing, operational use, or design specifications.

Entity-level movement models are typically table- or parameter-driven, where a function of entity type and terrain surface type determine maximum speed. Entity performance parameters such as turn radius and acceleration may be considered.

Entity direct fire combat models typically use conditional probability tables that encode the probability of a hit (P_h) given a shot and the probability of a kill given a hit (P_k). A simple P_h table might have two dimensions, weapon system and range; more sophisticated models will include other dimensions, e.g., target velocity or target aspect. The table entries are probabilities; e.g., a hit is scored if a random number is less than the appropriate P_h table entry. (These tables can be combined into a single table giving the probability of a kill given a shot P_{ks}.)

Entity sensing at the entity level often revolves around the determination of intervisibility or "line of sight" (LOS); here the question is whether the LOS between two entities is blocked by intervening terrain. Specific algorithms for determining intervisibility vary widely. A simple LOS algorithm may extend single a ray measured from a single sensor point to a single target point, consider only terrain obstructions, and return only "blocked" or "unblocked" results (Youngren 1994). More complex LOS algorithm extend multiple rays from the sensor point to multiple points on the target, consider not only the terrain but intervening physical obstructions such as vehicles, buildings, or trees having various opacities, and return a numerical value indicating what fraction of the target is potentially visible (CCTT 2003) (DISAF 2002) (OTB 2001). In any case, the specifics of most LOS algorithms are highly dependent on the representation format of the terrain database. It is important to note that LOS determinations can be computationally expensive; research to reduce the cost, both algorithmically and heuristically, has been on-going (Petty 1992) (Rajput 1995) (Petty 1997a) (Petty 1997b).

TECHNICAL APPROACH

This section describes the technical approach used for the new aggregate combat resolution algorithm.

Overview

The goal is to develop a new alternative aggregate (unit-level) combat model for moving, sensing, and shooting that can replace (or augment) current aggregate combat algorithms, such as Lanchester equations. Like the existing aggregate algorithms, the new algorithms will resolve combat at the unit level, but unlike the existing algorithms, they will be based on entity-level models. The development faces two challenges: (1) to preserve the detail and data support of the entity-level models in an aggregate model that is not so computationally expensive as to preclude its use in a unit-level simulation; and (2) to do so in a way that is mathematically sound.

We do so by abstracting away computationally expensive details yet retaining the essential effects of the entity-level models. Our approach will be to convert the detailed entity-level models into probability functions that are based on entity-level models but do not require fully detailed entity-level calculation at execution time. In essence, the details of the entity-level models will be "rolled up" into a small set of probability functions that will be embedded in a procedure for resolving unit-to-unit engagements.

The basic project goal is an alternative aggregate (unit-level) combat resolution algorithm that is based on entity-level performance data. That combat resolution algorithm should in some way "roll up" entity-level capabilities into unit-level effects for use in combat resolution, avoiding an entity-by-entity time-stepped pseudo-disaggregation-style approach.

The following had been completed at the end of the first phase of work on the project:

1. Developed a mathematical approach to resolving unit level combat that addresses the project goals.
2. Completed the development of initial versions of the probability functions for intervisibility, spotting, hit-and-kill, and location contained within the "kills" formula. Those initial versions are based on the terrain surrounding Bastogne Belgium (for intervisibility), the Panzer® miniatures rules for WWII entity-level ground combat (for spotting and hit-and-kill), and doctrinal unit formations (for location).
3. Implemented the "kills" formula and component probability functions in an executable form as a simple program (simple meaning without graphic interfaces, extensive input/output capabilities, and so on).
4. Prepared a test scenario based on a historical battle of approximately battalion size drawn from the 1944 Battle of the Ardennes, in particular the fighting around Bastogne, for simulation using the new combat resolution approach. This preparation will include historical order of battle and losses.
5. Simulated the test scenario using the executable form of the "kills" formula and component functions, compared the calculated results with the historical results, and analyzed the differences.

Combat resolution equations

The main equations of the combat model are shown in Figure 1. K_B is the number of kills inflicted by the Blue force on the Red force in a round of combat. The two summations in K_B are over all possible entity types (e.g., M4 Sherman). Factors included in the equation are number of entities of the type, rate of fire of that entity type, and duration of the round of combat. The KB portion of the equation computes, for each pair of opposing entity types, how many kills the entities of the attacking type score on the entities of the defending type.

Z_B, which is a factor in the K_B calculation, is the probability that a single entity of a given type kills a single entity of an opposing type in a single shot. Z_B is based on four probability functions that embody the detailed entity-level data that is the basis of the model. Those probability functions are described in the following sections.

A battle between two units is divided into rounds of combat. In each round, the basic equations are used to compute the losses inflicted by each force (Blue, Red) on the other and the losses are applied. The rounds repeat until a suitable end of engagement condition is satisfied (e.g., one of the forces is sufficiently reduced in strength, or a certain amount of time has passed).

$$K_B = \sum_{e_B \in E_B} \sum_{e_R \in \boldsymbol{E}_R} n_B(e_B) \cdot f_5(e_B) / t \cdot n_R(e_R) / n_R \cdot Z_B$$

where

$$Z_B = \int_{A_B} \int_{A_R} f_4(u_B, e_B, x_B, y_B) f_4(p_R, e_R, x_R, y_R) f_2(e_B, e_R, r, f_1(a, r) f_3(e_B, e_R, r) dA_R dA_B$$

and

$\boldsymbol{B}, \boldsymbol{R}$	= Blue (Red)
$E_{\boldsymbol{B}}, E_{\boldsymbol{R}}$	= Set of all Blue (Red) entity types (not just those in the engaged units)
B, R	= Blue (Red) unit in the engagement
u_B, u_R	= Unit information (unit type, center, facing, formation) for Blue (Red) unit
A_B, A_R	= Terrain area in which entities of Blue unit B (Red unit R) may be found
e	= Type of entity
$n_B(e), n_R(e)$	= Number of entities of type e in Blue unit B (Red unit R)
n_B, n_R	= Number of entities of all types in Blue unit B (Red unit R)
r	= Euclidian distance $((x_1 - x_2)^2 + (y_1 - y_2)^2)^{1/2}$
t	= Duration (minutes) of round of combat
$f_1(a, r)$	= Probability of intervisibility for terrain a at range r
$f_2(e_1, e_2, r, f_1(a, r))$	= Probability of detection from entity type e_1 to entity type e_2 at range r with intervisibility probability f_1
$f_3(e_1, e_2, r)$	= Probability of kill from entity type e_1 to entity type e_2 at range r (includes Pk and Ph)
$f_4(u, e, x, y)$	= Probability in a unit u of entity type e being located at x, y
$f_5(e)$	= Rate of file (shots/minute) of entity of type e

Figure 1. Main combat model equations.

Probability function: intervisibility

Terrain affects entity-level combat to a significant degree, due in large part to intervisibility; intervening terrain may block line of sight between pairs of hostile entities and prevent them from engaging via direct fire. Explicit entity-level intervisibility determination can be computationally expensive, and doing so for all pairs of hostile entities in two units is not practical in an aggregate-level algorithm. We would like to abstract away the details of both terrain and intervisibility, so as to reduce the computational burden, without losing the important effects they can have. The effect terrain has on entity-level combat can be modeled implicitly without explicitly representing the terrain itself or performing individual intervisibility determinations. That implicit representation of terrain and its effect on intervisibility is via the following probability function:

f_1 (a, r) = intervisibility value between two entities in terrain area a at range r

Function f_1 returns numerical intervisibility values ($0 \leq f_1$ $(a, r) \leq 1$, where is 0 is completely blocked 1 is completely unblocked, and other values represent partial intervisibility). Its return value is generated probabilistically, reflecting the typical range of intervisibility values for terrain area a at range r, so multiple calls to f_1 with the same parameters will not necessarily return the same value. (In strict mathematical terms, the fact that f_1, as well the other probability functions to follow, might produce different outputs for the same inputs means that they are not functions. We are certainly aware of this but will use the term regardless as it is suggestive of the purpose.) It is terrain specific, i.e., the range and distribution of values returned by f_1 will depend on the terrain area, as would be expected.

Function f_1 (or more precisely, the distribution parameters used by f_1) will be generated off-line in advance of simulation execution by automated analysis of the unit-level simulation's terrain. The procedure to be used is relatively straightforward. For each terrain area (the size of which would depend on the terrain database's size and variability), a grid (square or triangular) of points regularly spaced over the terrain and located a typical sensor height (e.g., 2 m) above the terrain will be generated. For each pair of points the range and the intervisibility value between them will be determined. This process will produce a set of data pairs of the form (range, intervisibility value). This data set will be analyzed using an averaging technique to produce distribution parameters (mean and standard deviation) for the intervisibility values found in the terrain area as functions of range. Function f_1, given a range, will use these functions to find the appropriate distribution parameters for the range and terrain, and then will use a random number and those parameters to generate a normally distributed intervisibility value.

It may be advantageous to partition the unit-level simulation's terrain into terrain areas that overlap so as to avoid edge effects or fair fight anomalies when entities are in different terrain areas. Also, if calculating the intervisibility function's distribution parameters for specific terrain databases is impractical, suitable distribution parameters for standard military terrain categories (open, mixed, closed) can be produced instead. Note that once the terrain has been analyzed it is no longer explicitly represented. Abstracting away the details and processing cost of the terrain while implicitly retaining its effects on the combat is a design intent.

It is possible to build f_1 automatically from terrain data sources. The first step for Bastogne was to obtain an authoritative terrain data set. We obtained a DTED Level 0 database of a 100km x 100km area centered on Bastogne. The next step is to sample this terrain data at various ranges (using multiple pairs of points at each range) to determine a percentage of unblocked lines of sight at that range. This becomes the value that we use for the range for f_1. For each of the 41 different ranges in Panzer, we computed 200,000 random pairs of points with that range. We then calculated the number of those pairs which had an unblocked line of sight. The results (in Figure 2) demonstrate the expected drop in probabilities.

Probability function: sensing

Entities will have one or more sensor systems capable of detecting hostile entities. These sensor systems may or may not require an unblocked line of sight; e.g., crew vision probably would, audio detectors might not. The performance of the sensor systems for the entities represented in the unit-level model will be modeled by the following probability function:

> f_2 (e_1, e_2, r, f_1 (a, r)) = probability an entity of type e_1 detects an entity of type e_2 at range r given the intervisibility status between them returned by intervisibility function f_1.

Here, "detects" means the threshold at which a sighting entity has sufficient information about the location and identity of another entity to engage it in direct fire. As with f_1, the return value for f_2 is generated probabilistically, reflecting the typical range of detection likelihood; multiple calls to f_2 with the same parameters will not necessarily return the same value.

The distribution parameters used by function f_2 will be generated off-line in advance of simulation execution by analysis of entity-level sensor models and sensor performance data, either by manual analysis of the models and data, or by automated execution of the models over an appropriate set of input values. Function f_1, given entity types, a range, and an intervisibility value, will find the correct distribution parameters, and will use a random number against those parameters to generate a normally distributed intervisibility value.

Function f_2 describes how combatants spot each other. The spotting procedure described in the Panzer Miniatures rules has been implemented. Spotting defines a range in which the spotter can see adversaries modified by variables visibility, unit size, and terrain cover.

Probability function: shooting

Entities which have detected a hostile entity may seek to engage it via direct fire. As noted earlier, entity-level direct fire models are typically tables giving P_h (probability of hit) and P_k (probability of kill), based on operational test or field data. The details of these tables will be represented implicitly by the following probability function:

> f_3 (e_1, e_2, r) = probability of kill for entity type e_1 against entity type e_2 at range r

Some entities have multiple weapon systems; function f_3 will assume that the firing entity uses the best weapon for the target entity. Note that probability of hit P_h, probability of kill P_k, and minimum and maximum weapon ranges are all represented in this single function, i.e., the probability of kill returned by f_3 includes both the possibility of missing the target (P_h) and of not killing it if a hit is scored (P_k), and f_3 will returns 0 if the target is outside of weapon range.

As with f_1 and f_2, the distribution parameters used by function f_3 will be produced off-line in advance of simulation execution. Given P_h and P_k (or P_{ks}) tables for the entity types to be represented, the data values of those tables can be converted into a probability function with an appropriate surface fitting technique, such as least squares fitting.

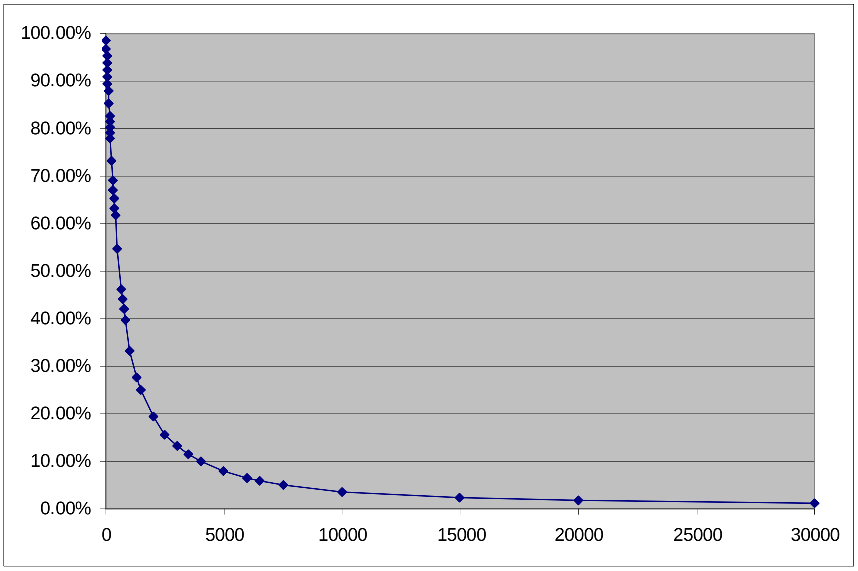

Figure 2. Values for f_1, computed from DTED0, up to 30,000m range.

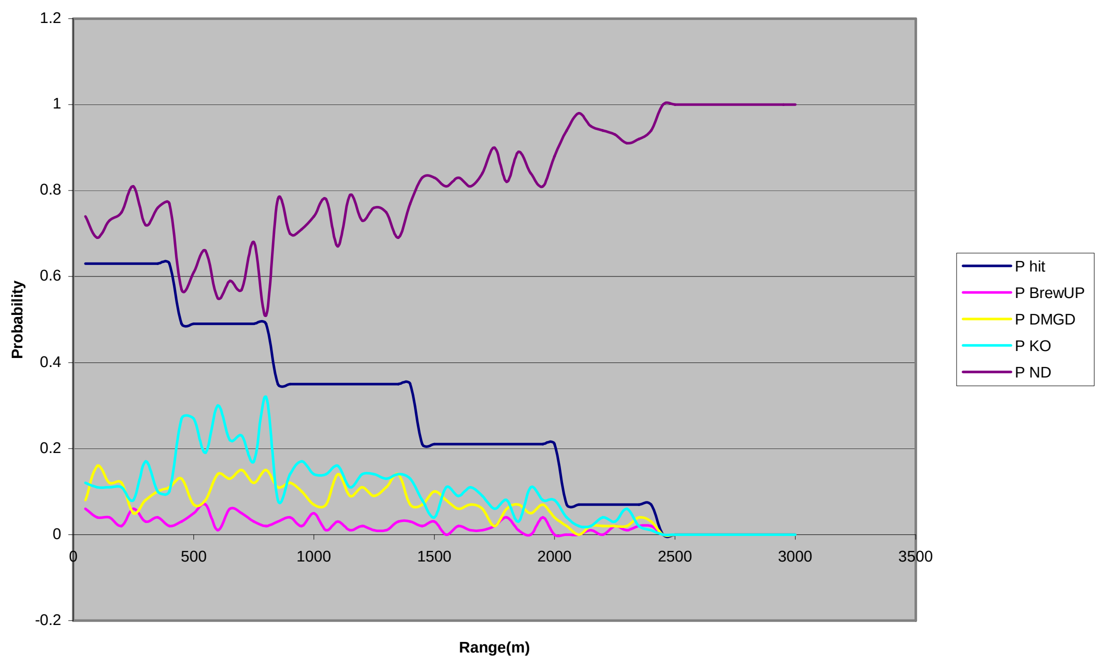

Figure 3. Example of f_3 probabilities derived from Panzer® rules.

Function f_3 encodes the Panzer® Miniatures rules, procedures and data and summarizes the probability that a unit of type A will hit and damage a unit of type B as a probability curve over a specific range. Since the data is different of each type of combatant, a different probability curve must be generated for each combination of combating vehicles.

The probability curves were generated by: 1) placing combatants facing each other at 50 meter increasing increments (up to the maximum possible range of effectiveness), and; 2) simulating a single round of combat (fire once, no return fire – depending on the type of weapon, multiple hits are possible with one round of fire). The data-generating program is tuned for combat around Bastogne, Belgium. This means that prevalent terrain conditions of this region, wooded cover, are used to modify the probability of hit. The program has accumulated data for each engagement for 100 tries as the Panzer miniature procedure involves several rolls of the dice thus giving a range of results.

The probability curves for each type of damage and the probability of hit are shown Figure 3. Note that BrewUp is symbolic in the game for the target catching on fire. Likewise, DMGD stands for "damaged", KO for knocked-out, and ND for no damage.

Several hit modifying conditions available to the Panzer Miniatures have not been used in the accumulation of the curves. For example, the combatants have been set up to face each other. This implies that all hits, if made, will be on the front part of the vehicle's armor. This side gives the most armor protection to the targeted vehicles. As a point of reference, the generating program was executed again, with both combatants facing in the same direction thus causing hits to impact on the rear armor of the target.

Probability function: location

The intent of f_4 is to represent the fact that certain types of entities are more likely to be closer (or further away) from the enemy, and thus more (or less) likely to kill or be killed by enemy fire, without having to explicitly represent the specific locations of the individual entities of the unit. For each unit type, formation, and entity type combination, standard unit formation diagrams will be used to generate a set of "formation points". Each formation point corresponds to the location where an entity of that type is expected to be located for a unit of that type in that formation. Function f_4 bases its probability calculation on the assumption that an entity is more likely to be nearer to the appropriate formation points, though not guaranteed to be precisely at them.

TESTING

The new combat resolution algorithm was tested by comparing its calculated results with the actual outcome of a historical battle from World War II. This section presents the testing process and results.

Historical battle

During the Battle of the Ardennes ("Battle of the Bulge"), as German forces were encircling Bastogne, U. S. Army forces held the village of Noville, located approximately 7 kilometers northeast of Bastogne along a main road that led to Houffalize, against German attacks (Cole 1965). The U. S. forces consisted primarily of the 1st Battalion of the 506th Parachute Infantry Regiment from the 101st Airborne and Team Desobry, a combined arms team of approximately two companies from the 10th Armored Division. The German attackers were from the 2d Panzer Division, especially the 3d Panzer Regiment and the 2d Panzergrenadier Regiment.

Over the course of December 19-20 1944 five separate actions were fought in defense of the village, the first coming in the early hours of December 19 and the last in the late afternoon of December 20. Much of the fighting took place under conditions of heavy fog which would appear and disappear over the course of the battle. The U. S. forces were eventually forced to withdraw from Noville and fight their way back to Bastogne, but not before delaying the encirclement of Bastogne by almost two days and inflicting heavy losses on the 2d Panzer Division.

The defense of Noville was selected because it was a reasonable size in terms of duration and engaged forces (large enough to be a reasonable test case, but not so large as to be difficult to analyze) and because neither indirect fire nor air support, which are not yet included in the model, had a significant impact on the battle.

Data preparation

An official U. S. Army history was used as the primary source regarding the defense of Noville (Cole 1965); information needed for the testing was developed from that source. Supplemental sources were consulted as well for information on standard unit organizations. Detailed and highly specific data was needed for each of the five actions, including order of battle (both sides were reinforced during the battle), weather conditions, unit deployments, engagement ranges, and losses. Because the primary source document is a historical narrative rather than a quantitative operations research analysis of the battle, not all the desired details were available; for example, specific losses for some of the actions were omitted or given in general terms "very heavy" or only for the overall Noville battle.

Nevertheless, close examination of the primary source combined with judicious application of knowledge of World War II combat permitted the development of detailed information regarding the defense of Noville to serve as comparison data for testing. For each of the five actions, the test data included the items listed earlier.

Table 1. Algorithm test results.

Action	Run 1, 2008-5-2		Run 2, 2008-5-5		Run 3, 2008-5-6		Run 4, 2008-5-7		Run 5, 2008-5-7	
	U. S. losses	**German losses**	**U. S. losses**	**German losses**	**U. S. losses**	**German losses**	**U. S. losses**	**German losses**	**U. S. losses**	**German losses**
(1)	0-0-0-0 0	0-0-0-0 0	0-0-0-0 0	0-0-0-0 0	0-0-0-0 0	0-0-0-0 0	0-0-0-0 0	0-0-0-0 0	0-0-0-0 0	0-0-0-0 0
(2)	1-3-0-0 4	2-1-0-0 3	2-10-3-2 17	4-2-0-0 6	1-8-0-1 10	3-2-0-0 5	0-2-0-0 2	0-1-0-0 1	0-2-0-0 2	0-1-0-0 1
(3)	0-0-0-0 0	0-8-0-0 8	13-6-18-2 39	13-16-3-0 32	11-9-15-3 38	5-13-4-0 22	1-4-7-0 12	0-4-2-0 6	1-3-5-0 9	0-4-2-0 6
(4)	0-0-0-0 0	0-12-0-0 12	4-3-7-2 16	5-9-4-0 18	6-3-13-2 24	4-8-4-0 16	1-3-10-1 15	1-4-3-0 8	0-2-8-0 10	0-9-5-0 14
(5)	14-4-0-0 18	0-9-0-0 9	3-0-26-6 35	1-3-3-0 7	3-0-27-5 35	1-3-3-0 7	3-4-17-5 29	1-18-12-0 31	3-3-10-4 20	1-18-9-0 28
Total	15-7-0-0 **22**	2-30-0-0 **32**	22-19-54-12 **107**	23-30-10-0 **63**	21-20-55-11 **107**	13-26-11-0 **50**	5-13-34-6 **58**	2-27-17-0 **46**	4-10-23-4 **41**	1-32-16-0 **49**
Expected	17-9-26-2 **54**	31-20-20-0 **71**	17-9-26-2 **54**	31-20-20-0 **71**	17-9-26-2 **54**	31-20-20-0 **71**	17-9-26-2 **54**	31-20-20-0 **71**	17-9-26-2 **54**	31-20-20-0 **71**
Total – Expected	-32	-39	53	-8	53	-21	4	-15	-13	-22

Test runs

The new combat resolution algorithm was used to model the defense of Noville. A total of five complete runs of the algorithm were made, each of which included all five of the separate engagements that made up the battle. Between each run, the results were compared to the historical outcome and adjustments were made with the intent of improving the correlation.

The inter-run adjustments fell into two categories, changes to the algorithm and corrections to the test data. Changes to the algorithm made as a result of the testing including modifying the shooting probability function (f_3) to reflect that fact that not all hits were scored on the front armor of target vehicles and modifying the sensing probability function (f_2) to take the fog prevalent in the Noville battle into account. Adjustments were made in the test data (e.g., changes to engagement ranges or action durations) when the run results indicated that estimates and informed guesses made in the initial analysis of the historical source may have been incorrect. Test data adjustments were made only to values absent or vaguely stated in the historical source; all explicit data values in the historical source were used exactly as given.

Test results

Table 1 summarizes the test results. The five test runs are shown vertically in the table, with separate columns for U. S. and German losses. The five actions that made up the defense of Noville are shown horizontally the table. Each cell reports losses for a specific action in a test run. The first row of numbers gives losses in four categories: tanks (including assault guns), other vehicles, infantry squads, and other non-vehicles (e.g., anti-tank guns). The single number in the second row is the total of all losses.

The last three rows of the table compare the expected results (i.e., the historical losses, as best as could be ascertained from the source) and the algorithm test results. The row labeled "Total" gives the totals losses for the overall battle in the test run. The row labeled "Expected" gives the historical losses; of course, these values do not change from test run to test run. The last row gives the difference between the test run losses and the historical losses.

While the test run losses and the historical losses do not match exactly for any of the test runs, the differences do get generally smaller over the test run sequence as the algorithm was adjusted, and they are within approximately 30% of the historical losses in the final test run, a satisfactory outcome for a new algorithm in its initial tests. Moreover, by the last test run the U. S. losses were less than the German losses, just as occurred historically, in spite of the superior numbers of engaged German forces, suggesting that the algorithm was properly taking into account factors that influenced the outcome, such as weather.

FUTURE WORK

Work is beginning on the second phase of this project. During the second phase, we plan to complete the following:

1. *Expand coverage of conventional warfare.* Expand the range of combat that the new alternative aggregate algorithm can model, in

terms of historical periods and combatant forces, for conventional kinetic warfare.

2. *Validate using historical data.* Validate the new alternative aggregate algorithm by comparing its calculated results with actual results from representative historical battles taken from a variety of time periods, from World War II to modern day.
3. *Compare results and performance with existing models.* Compare the new alternative aggregate algorithm with existing combat models, in terms of both battle outcomes and computational cost.
4. *Implement as software product.* Implement the new alternative aggregate algorithm as a software product, suitable for distribution, reuse, and integration with existing combat simulations.
5. *Integrate with existing simulation.* Integrate the software implementation of the new alternative aggregate algorithm with the U. S. Army's OneSAF simulation in such a way as to allow it to resolve combat that might otherwise have been resolved at the entity level or using OneSAF's current aggregate level combat resolution algorithm.
6. *Publish and promulgate.* Write and publish in appropriate venues scientific papers detailing all aspects of the new algorithm and the validation results. Present the algorithm and its results in suitable forums so as to elicit expert community feedback.
7. *Expand coverage to unconventional warfare.* Extend the base "kills" equation and component probability functions of the new alternative aggregate algorithm so as to enable it to model one or more types of combat or conflict categorized as unconventional, e.g., insurgent, asymmetric, terror, urban, non-kinetic, or PMESII.

ACKNOWLEDGEMENTS

This project was sponsored by the Defense Advanced Research Projects Agency under STTR ST071-003, and was supervised by COL John "Buck" Surdu. That support is gratefully acknowledged.

REFERENCES

L. B. Anderson et al., "IDA Ground-Air Model I (IDAGAM I)", *Report R-199*, Institute for Defense Analysis, Alexandria VA.

S. Bonder, "The Lanchester Attrition-Rate Coefficient", *Operations Research*, Vol. 15, pp. 221-232.

S. Bonder and R. L. Farrell (Editors), "Development of Models for Defense Planning", *Report SRL 2147*, TR-70-2, AD 714 667, Systems Research Laboratory, University of Michigan, Ann Arbor MI.

Close Combat Tactical Trainer (CCTT), Version 8.4, Source code, 2003.

H. M. Cole, *United States Army in World War II, The European Theater of Operations, The Ardennes: Battle of the Bulge*, Office of the Chief of Military History, Department of the Army, Washington DC, 1965.

D. P. Dare and B. A. P. James, "The derivation of Some Parameters for a Corps/Division Model from a Battle Group Model," *Defense Operational Analysis Establishment*, West Byfleet U.K., M7120.

P. K. Davis. *Aggregation, Disaggregation, and the 3:1 Rule in Ground Combat*, RAND MR-638-AF/A/OSD 1995.

Dismounted Infantry Semi-Automated Forces (DISAF), Version 9.4, Source Code, 2002.

T. N. Dupuy, *Numbers, Prediction, and War: Using History to Evaluate Combat Factors and Predict the Outcome of Battles*, Dupuy Institute, Annandale VA.

T. N. Dupuy, *Attrition: Forecasting Battle Casualties and Equipment Losses in Modern War*, Nova Publications, Falls Church VA.

T. N. Dupuy, *Understanding War: History and Theory of Combat*, Nova Publications, Falls Church VA, 1998.

B. W. Fowler, *De Physica Beli: An Introduction to Lanchestrial Attrition Mechanics*, Part I. IIT Research Institute/DMSTTIAC, Report SOAR 96-03.

B. W. Fowler, *De Physica Beli: An Introduction to Lanchestrial Attrition Mechanics*, Part II. IIT Research Institute/DMSTTIAC, Report SOAR 96-03.

B. W. Fowler, *De Physica Beli: An Introduction to Lanchestrial Attrition Mechanics*, Part III. IIT Research Institute/DMSTTIAC, Report SOAR 96-03.

R. W. Franceschini and M. D. Petty, "Linking constructive and virtual simulation in DIS", in T. L. Clarke (Editor), *Distributed Interactive Simulation Systems for Simulation and Training in the Aerospace Environment*, SPIE Critical Reviews of Optical Science and Technology, Vol. CR58, SPIE Press, Bellingham WA, 1995, pp. 281-298.

R. W. Franceschini, *Correlation Error in Multiple Resolution Entity Simulation*, Ph.D. Dissertation, University of Central Florida, 1999.

R. L. Helmbold, "A Modification of Lanchester's Equations," *Operations Research*, Vol. 13.

E. P. Kerlin and R. H. Cole, “ATLAS: A Tactical, Logistical, and Air Simulation”, Technical Paper RAC-TP-338, Research Analysis Corporation.

B. O. Koopman, *Search and Screening*, Military Operations Research Society (MORS), Alexandria VA.

OneSAF Testbed Baseline (OTB), Version 1.0, Source code, 2001.

R. Peterson, “Methods of Tank Combat Analysis,” *Report of the Fifth Tank Conference*, edited by H. Goldman and G. Zeller, Ballistic Research Laboratory, Aberdeen Proving Grounds, MD, Rept. 918, AD 46000.

M. D. Petty, C. E. Campbell, R. W. Franceschini, and M. H. Provost, “Efficient Line of Sight Determination in Polygonal Terrain”, *Proceedings of the 1992 IMAGE VI Conference*, Phoenix AZ, July 14-17 1992, pp. 239-253.

M. D. Petty, *Computational Geometry Techniques for Terrain Reasoning and Data Distribution Problems in Distributed Battlefield Simulation*, Ph.D. Dissertation, University of Central Florida, 1997.

M. D. Petty and A. Mukherjee, “The Sieve Overlap Algorithm for Intervisibility Determination”, *Proceedings of the 1997 Spring Simulation Interoperability Workshop*, Orlando FL, March 3-7 1997, pp. 245-255.

M. D. Petty and R. W. Franceschini, *Interactions Across the Aggregate/Disaggregate Simulation Boundary in Multi-Resolution Simulation*, Technical Report, Institute for Simulation and Training, December 20 1998.

S. Rajput, C. R. Karr, M. D. Petty, and M. A. Craft, “Intervisibility Heuristics for Computer Generated Forces”, *Proceedings of the Fifth Conference on Computer Generated Forces and Behavioral Representation*, Orlando FL, May 9-11 1995, pp. 451-464.

J. T. Taylor, *Lanchester Models of Warfare*, Vol. I, Defense Technological Information Center (DTIC), ADA090843, Naval Post Graduate School, Monterey CA.

J. T. Taylor, *Lanchester Models of Warfare*, Vol. II, Defense Technological Information Center (DTIC), ADA090843, Naval Post Graduate School, Monterey CA.

J. T. Taylor, *Force-on-Force Attrition Modeling*, Military Operations Research Society (MORS), Alexandria VA.

A. R. Washburn, *Search and Detection, 4th Edition*, Operations Research Section, INFORMS, Baltimore MD.

M. A. Youngren, *Military Operations Research Analyst’s Handbook, Volume I: Terrain, Unit Movement, and Environment*, Military Operations Research Society (MORS).