

EINStein Goes to War

A Primer on Ground Combat Models

Christopher P. Fredlake and Kai Wang

CIM D0018865.A1/Final
September 2008

Approved for distribution:

September 2008



Dr. Delwyn Gilmore
Tactical Naval Issues
Operations and Tactics Analysis

This document represents the best opinion of CNA at the time of issue.
It does not necessarily represent the opinion of the Department of the Navy.

Distribution unlimited. Specific authority: N00014-05-D-0500.
Copies of this document can be obtained through the Defense Technical Information Center at www.dtic.mil
or contact CNA Document Control and Distribution Section at 703-824-2123.

Copyright © 2008 CNA

Contents

Summary	1
Introduction	3
Models and simulations defined.	4
History of modeling	6
Wargames in wood and stone	6
Lanchester-type equations	7
Computerized models	8
Survey of combat models	8
OneSAF	9
CASTFOREM	10
JCATS	10
Limitations of current models.	11
Lanchester-type attrition equations	13
Functional forms of Lanchester-type equations	13
The general form	13
The Square Law.	14
The Linear Law	15
The Mixed Law	17
The Logarithmic Law.	18
The Morse and Kimball case	19
Summary	20
Key assumptions	20
Agent-based models offer a new approach	21
“EINSTEIN” goes to war	23
Review of CNA ground combat studies	23
Modeling a wide range of scenarios	24
Defensive combat modeled in EINSTEIN	26
Combat in open terrain and urban warfare	28
Discussion of the use of EINSTEIN for the GCS.	30
Comments on VV&A	33

Summary	34
Basic setup for using EINSTEIn	35
Setting up the battlefield and forces	36
Assigning squad personalities	36
Weapon arsenals and assignments	40
Movement and sensor ranges	41
Terrain elements	41
Examples of battle simulations using EINSTEIn	42
Example 1: Linear battle fronts	42
Example 2: Dispersive battle	45
Example 3: Swarming Blue forces	46
Example 4: Red feint retreat	49
Adding terrain elements	51
Examples of other CA-based models	52
Conclusions	53
Glossary	55
References	57
List of figures	61
List of tables	63
Distribution list	65

Summary

In this CNA Information Memorandum (CIM), we present a primer on ground combat models with brief reviews on Lanchester-type attrition equations and how CNA used a multiagent-based model, EINSTEIn¹, to study small-unit ground combat tactics.

The U.S. Department of Defense (DoD) uses many types of models in applications ranging from acquisition, weapon system design, test and evaluation, training, tactics development, etc. However, it is widely accepted that “all models are wrong, and some are useful.” This can be mostly attributed to the emphasis of many current combat models on modeling software and computer hardware while neglecting the proper application of fundamental military science [1]. For example, many combat models incorrectly use Lanchester-type equations to calculate force-on-force attrition when the assumptions of the Lanchester equations are not appropriate for the models.

Compared to many conventional deterministic models, agent-based models offer a new approach to improve our understanding of combat. By specifying a set of simple inter-agent relationships, one can access complex behavior space that emerges from multiple runs of an agent-based model and gain insights not available through the conventional models.

Recent CNA ground combat studies using EINSTEIn examined a wide range of small-unit scenarios and showed EINSTEIn to be an efficient and effective analysis tool. For interested readers, we provide a quick guide to EINSTEIn and walk through several examples to show the various features of the software toolkit.

1. EINSTEIn refers to Enhanced ISAAC Neural Simulation Toolkit, and ISAAC stands for Irreducible Semi-Autonomous Adaptive Combat.

EINSTEIN and its predecessor ISAAC have opened a new chapter in military operations research and have inspired many other models based on the same mathematical foundation. As the tools improve and evolve, they will provide novel approaches for a better understanding of the complex dynamics of combat.

Introduction

Many readers may be familiar with classic works on warfare, such as *The Art of War* by Sun Tzu in sixth century B.C. China and Carl von Clausewitz's *On War* in 19th century Prussia. For as long as there has been warfare, humans have tried to understand the essence of this complex social phenomenon. Today, practitioners of military sciences continue to search for ways to better understand the complex dynamics on a battlefield and improve their chances against potential foes.

This has led to the birth of a new branch of military operations research, the application of complexity theories to ground combat models. In the forward section of the book *Artificial War* by Andrew Ilachinski, retired Marine Corps LtGen Paul K. Van Riper recalls the Marine Corps's role in this effort[2]:

The United States Marine Corps in 1994 undertook a wide review of new discoveries seeking those that showed promise for improving the profession of arms. 'Casting their nets widely' and looking far beyond the usual interest of military personnel, a handful of Marine officers—of which I was fortunate to be included—learned of the emerging field of nonlinear dynamics, more popularly known as the science of chaos or complexity.

Soon after LtGen Van Riper took over as the Commanding General of the Marine Corps Combat Development Command (MCCDC) in 1995, he established the Office of New Sciences to explore military applications of novel scientific theories. He eventually approached Andrew Ilachinski at CNA to conduct "research on the relevance of complexity theory to land combat." The result of that collaboration was the groundbreaking multiagent-based model Irreducible Semi-Autonomous Adaptive Combat (ISAAC). Encouraged by the preliminary success of ISAAC, Ilachinski developed the Windows-based EINSTEIN (or Enhanced ISAAC Neural Simulation Toolkit). The software toolkit has been continuously updated over the years and used in a series of CNA studies on ground combat.

In this CNA information memorandum (CIM), we provide a primer on ground combat models starting with a discussion on the various different types of models. We then briefly review a few combat models currently used by the DoD community. Because many of these models use Lanchester-type equations in their attrition calculations, we explore a few examples of these equations. However, because of the inherent limitations of the Lanchester-type equations, they are ill-equipped to model many real-world scenarios. In contrast, multi-agent-based models like EINSTEIN have been shown to offer novel approaches to better understand warfare. We also review a recent series of CNA ground combat studies (GCS) using EINSTEIN² and provide a primer on using EINSTEIN along with some sample scenarios for interested readers to get started quickly on using the software.

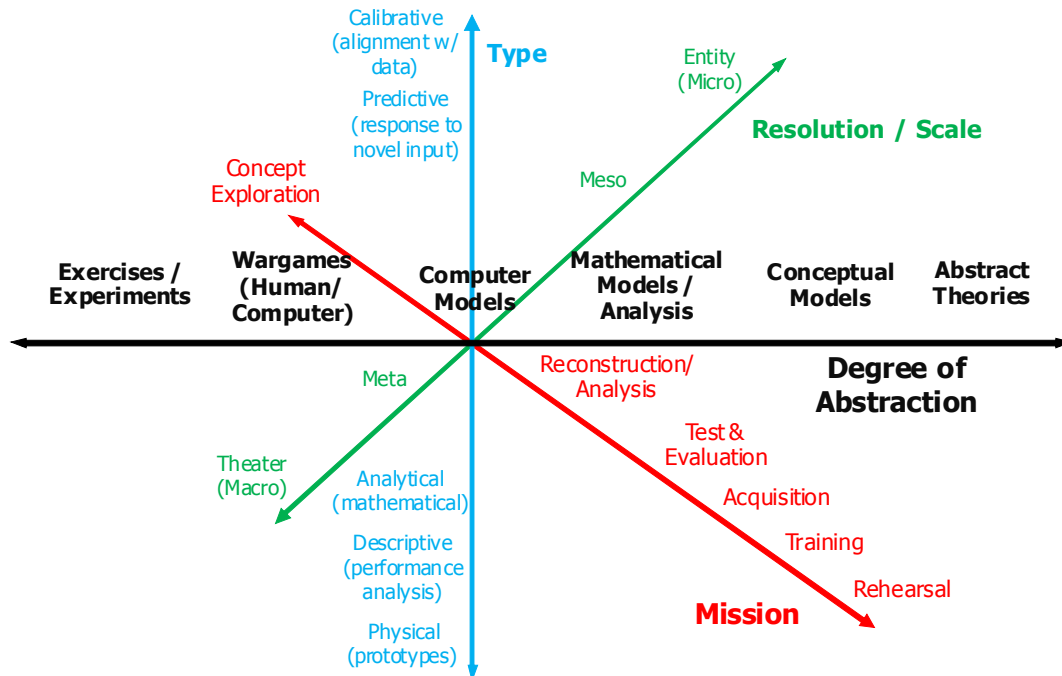
Models and simulations defined

The terms “model” and “simulation” are often used interchangeably. However, DoD presents much stricter definitions that differentiate the two terms. According to [3], a model is defined as “a physical, mathematical, or otherwise logical representation of a system, entity, phenomenon, or process.” A simulation is “a method for implementing a model over time.” Modeling and simulation (M&S) are defined as “the use of models, including emulators, prototypes, simulators, either statically or over time, to develop data as a basis for making managerial or technical decisions.” The key difference between models and simulations in the DoD context is the time variable. One may think of a picture as a model, where elements of the picture depict some real-world objects. A series of pictures played over time (i.e., a simulation) tells a story about these objects.

Figure 1 shows the various types of modeling methods and how they may be classified.

2. We focus on the period from 2003 to 2006. Ilachinski summarized prior CNA work from 1996 to 2003 in his book *Artificial War* [2].

Figure 1. Various dimensions to modeling^a



a. Source: Andrew Ilachinski. Modeling and Simulation: An Analyst's Perspective (The Good, Bad, Chaotic, and Complex), Aug 2006 (CME D0014738.A1)

While real-world warfare is often classified as either defensive or offensive [4], models are rarely defined in such terms. Models shown in figure 1 are based on a much broader definition than the DoD version. They can be classified according to their:

- Degree of abstraction (e.g, from field exercises to abstract theories and analytical equations, shown in black in figure 1)
- Resolution and scale (e.g., from an individual soldier to an entire army, shown in green in figure 1)
- Type (e.g., deterministic vs. stochastic or logical process models vs. artificial intelligence models, shown in blue in figure 1)
- Mission (e.g., design, acquisition, tactics development, training, etc., shown in red in figure 1).

In terms of the degree of abstraction of models, the most concrete (and least abstract) are military field exercises and experiments. They provide the most realistic simulations for real-world operations. However, they provide limited opportunities for analysis due to high costs³ and various artificialities.

At the other end of the spectrum, an analyst can use a set of analytical equations such as the Lanchester-type equations to quickly estimate attrition numbers for real-world operations; however, these equations are often too abstract and have too many simplifying assumptions to provide realistic results.

Computerized models fall somewhere in the middle. Most DoD models and multiagent-based models belong to this category, and they will be explored in more detail later.

History of modeling

We present here an abbreviated history of combat models. Others have written extensively on this subject, and we recommend that interested readers check out *The Art of Wargaming* by Peter Perla and Andrew Wilson's *The Bomb and the Computer* [6, 7]. When models are discussed, a computer model often comes to mind. However, long before computers were used, board game-style wargames, made of wood and stone, were invented to simulate land warfare [6]. We begin our discussion of the history of combat models with these earliest board games.

Wargames in wood and stone

The essence of warfare was captured in board games like the Indian game *Chaturanga* (the predecessor of today's chess) and *Wei Qi* in ancient China [6]. Both games are high abstractions of warfare and were played using fairly simple rules. Since most readers are probably familiar with chess, we elaborate more on the rules of *Wei Qi*.

3. In 2002, the three-week long exercise Millennium Challenge involved 13,500 participants and was estimated to cost \$250 million [5].

Wei Qi can be loosely translated to “game of encirclement”; its origin goes back over four thousand years [8]. Today, the game is widely played in East Asia. It is known as *Go* in Japan and *Baduk* in Korea. The game involves two players taking turns to place black or white stones over intersections of a 19 by 19 grid board. Stones are removed from the board when they are deemed to have been surrounded (i.e., “encirclement”) and can no longer “maneuver.” The player whose stones occupy the larger territory on the board at the end wins the game.

The simple rules of these games have remained unchanged for centuries, and these highly stylistic games probably had limited applications in helping players to understand combat.

This changed dramatically when in 1664, Christopher Weikmann invented the game *Koenigspiel*, or “King’s Game” (also known as military chess). Using chess as a basis, Weikmann added additional pieces and roles to the game, and he introduced a die to decide the outcome of engagements between opposing forces. *Koenigspiel* revolutionized boardgame-based wargames. For the first time, a game had enough detail and “hierarchy” to construct models for real-world scenarios. Weikmann remarked, perhaps slightly self-servingly, that the game “would furnish anyone who studied it properly a compendium of the most useful military and political principles” [6].

Lanchester-type equations

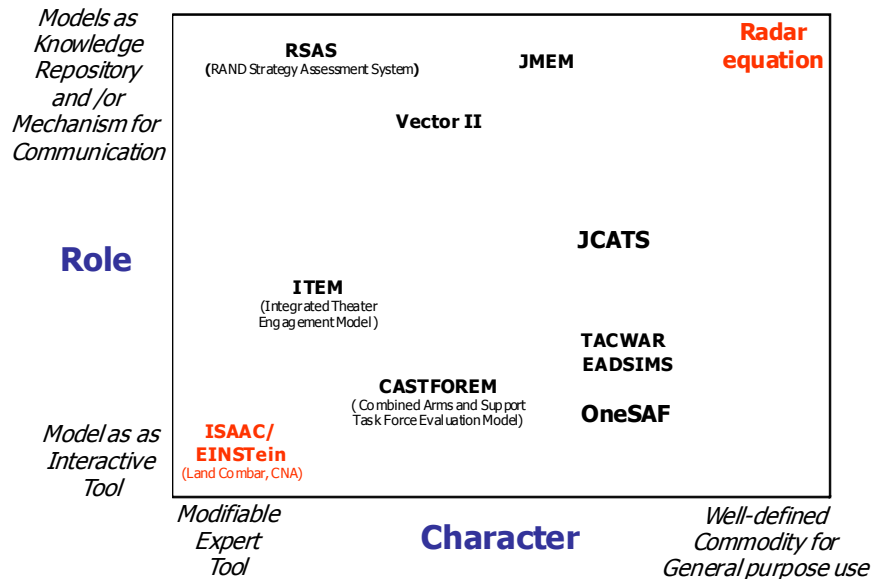
Developments in science and technology over the next two centuries created the foundation for the combat models we use today. The next major step in warfare modeling was developed by the British engineer, Frederick William Lanchester during World War I. Lanchester devised a series of differential equations to demonstrate the relationships between opposing forces in his publication *Aircraft in Warfare: The Dawn of the Fourth Arm* [9].⁴ Lanchester-type equations are still used in many of the modern computerized models, and they are discussed in more detail in the next chapter.

4. Similar mathematical formulations were put forth by others (Chase and Osipov) at about the same time.

Computerized models

Ever since computers were first used with military models during World War II, combat models have gained significant capability and capacity. Mathematical algorithms and computer databases have replaced wooden boards and dice from an earlier era. Echoing the earlier classification of models, figure 2 shows the different uses of models including various computer models [10]. We explore a few of these models in more detail in the following section.

Figure 2. Use of different models^a



a. Source: Andrew Ilachinski. Modeling and Simulation: An Analyst's Perspective (The Good, Bad, Chaotic, and Complex), Aug 2006 (CME D0014738.A1)

Survey of combat models

For a comprehensive survey of DoD models, we recommend a 1996 Air Force Research Laboratory report, *Warfare Simulations: Status and Issues for Space*, which was updated in 2003 [11, 12]. Information can also be found at the DoD Modeling and Simulation Coordination

Office (M&S CO⁵) as it is “the Executive Secretariat for DoD M&S Management in fostering the inter-operability, reuse, and affordability of crosscutting M&S to provide improved capabilities for DoD operations” [14].

In this chapter, we focus on three ground combat models in use today. They are One Semi-Automated Forces (OneSAF), the Combined Arms and Support Task Force Evaluation Model (CAST-FOREM), and the Joint Conflict and Tactical Simulation (JCATS).

We are limited to fairly abstract descriptions of these models as opposed to more detailed discussions on their algorithmic foundation. This is due to the limited time and scope of this report and the difficulty in getting access to the source documents of these models.

OneSAF

OneSAF is designed as

a composable, next generation, entity-level Computer Generated Forces (CGF) simulation designed for brigade and below, combat and non-combat operations. OneSAF has been crafted to model detailed buildings for urban operations including interior rooms, furniture, tunnels and subterranean features [15].

OneSAF models more than 25 sides and forces with asymmetric side relationships to better model the contemporary operating environment. As an example, in OneSAF it is possible to model two tribes that are both friendly to the Blue forces but are enemies to each other. OneSAF version 1.5, released on 23 August 2007, incorporates improvements in the following capabilities: Improvised Explosive Devices (IEDs), the creation of mouse holes, dynamic side changes, reduced profile shooting, urban medical evacuation, snipers, projectile penetration of interior walls, conduct raids and ambushes, shielding tactics, decoys, craters and naval operations [15].

5. M&S CO is the former Defense Modeling and Simulation Office (DMSO). Its Essentials of Modeling and Simulation (EMS) web portal is a good place to start [13].

CASTFOREM

The Combined Arms and Support Task Force Evaluation Model (CASTFOREM) is:

the Army's highest resolution, combined arms combat simulation model. CASTFOREM is used for weapon systems and tactics evaluation in brigade and below combined arms conflicts. The model uses closed-form mathematical expressions, probability distributions, along with an embedded expert system (which is implemented via decision look-up tables) to perform some elements of command and control. It was designed primarily to simulate intense firefights about 1 to 1-1/2 hours in duration for echelons up to including brigade. However, because individual scenarios take about as long to run as the firefights the program is simulating, it is difficult to conduct any meaningful exploratory analyses [2].

JCATS

The U.S. Army and Marine Corps have pioneered ground combat modeling since World War II. As warfare becomes more Joint, the M&S world is also becoming more Joint. For example, the Joint Conflict and Tactical Simulation (JCATS) is managed by the U.S. Joint Forces Command's Joint Warfighting Center and was initially developed by the Lawrence Livermore National Laboratory (LLNL). JCATS is

a multi-sided interactive, entity-level conflict simulation used by military and site security organizations as an exercise driver and a tool for training, analysis and mission planning. Among the unique capabilities provided by JCATS is very detailed modeling of small group tactics in rural or urban terrain modeling day or night operations with artificial lighting. JCATS also provides dynamic aggregation and de-aggregation of units during the game, allowing the user to control large numbers of entities with fewer players. Because it is a data-driven model, proper data configuration provides tremendous flexibility in modeling situations or equipment [16].

JCATS combines the ability to model large 600 kilometers by 600 kilometers play box scenarios with the detail needed to handle a hostage rescue situation inside a building. It is able to dynamically aggregate and de-aggregate units to an

entity level. This functionality greatly facilitates Distributed Interactive Simulation or High Level Architecture interactions with highly aggregated models such as the Joint Theater Level Simulation. JCATS differs from most other combat simulations primarily in its terrain and feature representations, detailed dismounted infantry and urban features and ability to simulate large joint exercises [16].

Limitations of current models

As we mentioned earlier, it was very difficult to get access to the source algorithmic documents of these models. In rare occasions where we were successful, we often found that the supporting documents were lacking in necessary detail. For example, we obtained a copy of the JCATS algorithm manual from the developers at Lawrence Livermore National Laboratory, but we found that eight out of the 15 chapters on the model algorithm were yet to be written, including the chapter on attrition calculations [17].

The lack of easily accessible documentation aside, Ilachinski summarizes that all of these current models “are very complex, are usually tied to a small set of specific computing hardware, are difficult to interface, have limited data collection facilities, and require real-time (or close to real-time) run times (thus making it effectively impossible to conduct meaningfully large exploratory analyses of possible behaviors)” [2].

Multiagent-based models like EINSTEIN provide a new class of models that are “smaller, more tightly-focused models, that can help with basic concept development and be used to enhance intuition about fundamental principles and expected behaviors” [2].

Because many DoD combat models rely on Lanchester-type equations for attrition calculations, we provide a more detailed presentation on Lanchester-type attrition equations in the next section followed by the discussion on EINSTEIN.

This page intentionally left blank.

Lanchester-type attrition equations

Ever since they were proposed during World War I, the Lanchester equations have found wide use and taken many different functional forms. Although they have simple mathematical formulations, Lanchester-type equations are quite capable of modeling various combat scenarios. A caveat is that the equations are based on many simplifying assumptions that are often not suitable for the problems they are asked to solve. In the next section, we show a few examples of Lanchester-type equations. More detailed presentations on this subject can be found in [18, 19].

Functional forms of Lanchester-type equations

The general form

The general form of Lanchester-type equations may be summarized as follows:

$$\frac{dR}{dt} = N_R - K_B \cdot B^r \cdot R^s - \alpha \cdot R \quad \frac{dB}{dt} = N_B - K_R \cdot R^v \cdot B^u - \beta \cdot B$$

In the equations above, the variables R and B represent the size of the Red force and the Blue force *actively engaged in a battle* at time t . N_R and N_B are positive values that represent the rates of additional Red and Blue troops that can be brought to the battle, respectively. Parameters r , s , v , and u are time-independent parameters. K_B and K_R are positive constants referred to as Lanchester attrition rate coefficients (e.g., K_B represents the effectiveness of Blue force firepower against the Red force). α and β are also positive constants referred to as operational loss rate coefficients.

By assigning values of 1 or 0 to r , s , v , u , α , and β , we can generate various functional forms of the Lanchester-type equations. In the

next section, we present five such functional forms including: the Square Law, the Linear Law, the Mixed Law, the Logarithmic Law, and the Morse and Kimball case.

In all but the last example, we assume $N_R = N_B = 0$.

The Square Law

The Square Law is derived from the general Lanchester-type equations by taking $r = v = 1$, $s = u = 0$ and $\alpha = \beta = 0$. We get the following differential equations:

$$\frac{dR}{dt} = -K_B \cdot B \qquad \frac{dB}{dt} = -K_R \cdot R$$

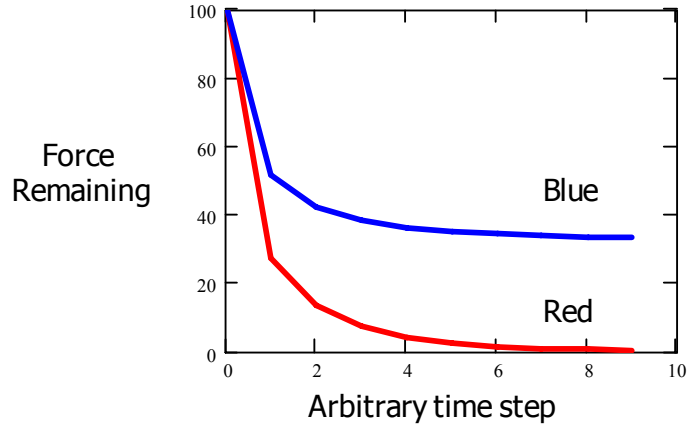
The Square Law takes its name from the second-order state equation shown below:

$$K_R \cdot (R_0^2 - R^2) = K_B \cdot (B_0^2 - B^2)$$

To illustrate, let us assume that both the initial Red force (R_0) and initial Blue force (B_0) sizes are 100 (or $R_0 = B_0 = 100$) and the ratio of K_B to K_R is 1.5 (i.e., the Blue force is 50 percent more lethal than the Red force). We plot the numerical solutions of the force strength (R and B) over artificial time steps using MathCAD.⁶ This is shown in figure 3.

6. The numerical solutions shown here are solved using the *rkfixed* subroutine in the MathCAD software. The *rkfixed* subroutine uses a fourth-order Runge-Kutta integration algorithm with a fixed integration step size to solve sets of ordinary differential equations.

Figure 3. Lanchester Square Law chart



The Square Law is often used to model the type of static trench warfare that was dominant during World War I. Attrition of the Red force is directly related to the Blue firepower (expressed as the product of the number of Blue shooters, B , and their weapons' effectiveness, K_B). In the example shown above, both forces start with the same number of troops. However, because the Blue force's weapon effectiveness is 50 percent greater than that of the Red force, Blue wins the battle.

The Linear Law

If we assume $r = v = s = u = 1$ and $\alpha = \beta = 0$ in the general equations, we get the following differential equations:

$$\frac{dR}{dt} = -K_B \cdot B \cdot R \qquad \frac{dB}{dt} = -K_R \cdot R \cdot B$$

or

$$\frac{1}{K_B} \cdot \frac{dR}{dt} = \frac{1}{K_R} \cdot \frac{dB}{dt}$$

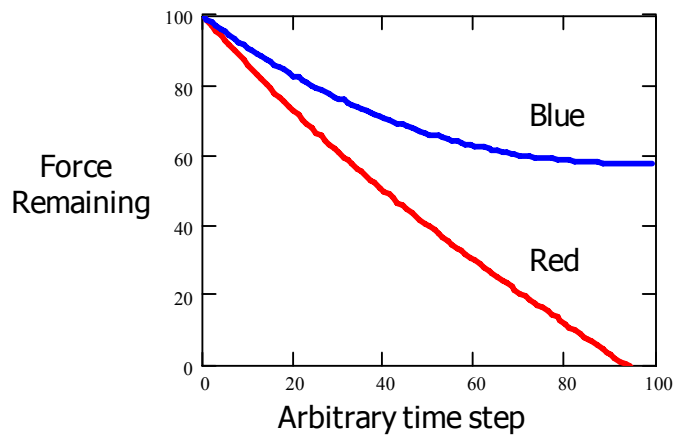
The mathematical statement above shows that the rate of loss for the forces is independent of the force size. They are determined by the attrition rate coefficients, K_B and K_R .

The Linear Law takes its name from the first-order state equation shown below:

$$K_R \cdot (R_o - R) = K_B \cdot (B_o - B)$$

Using the same initial conditions as for the Square Law case, we assume that $R_o = B_o = 100$ and K_B to K_R is 1.5 and plot the numerical solutions in figure 4.

Figure 4. Lanchester Linear Law chart



The Linear Law form has been thought to most closely simulate conditions of artillery fire on enemy positions or the hand-to-hand combat in close quarters. The sizes of the forces involved have no influence on the attrition rates. The only thing that matters is the effectiveness of the respective weapons used. As shown in figure 4, the Blue force has a clear advantage because of its superior firepower. By $t = 90$, most Red troops are eliminated, while roughly 40 percent of the Blue force is alive.

The Mixed Law

If we assume that $r = v = u = 1$, $s = 0$, and $\alpha = \beta = 0$, the generalized Lanchester-type equations become the equations shown below:

$$\frac{dR}{dt} = -K_B \cdot B \quad \frac{dB}{dt} = -K_R \cdot B \cdot R$$

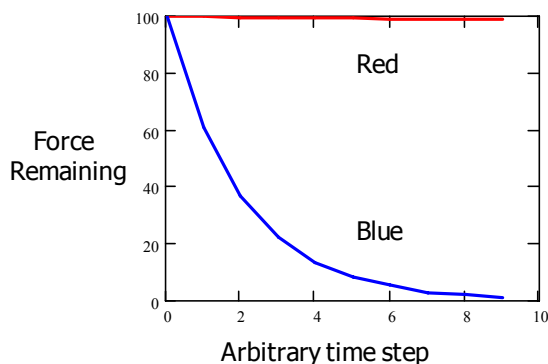
In this example, the Red force attrition equation is identical to the Square Law. The Blue force attrition, on the other hand, is adversely affected not only by the number of the Red shooters but also by the number of the Blue force (as targets). One can consider this as a scenario where the Blue force is being ambushed by the Red force (i.e., Red shooters can take aim at Blue targets more easily than the other way around).

The Mixed Law takes its name from the mixed first- and second-order state equation shown below:

$$\frac{K_R}{2} \cdot (R_o^2 - R^2) = K_B \cdot (B_o - B)$$

If we assume that both the Red and Blue forces have initial strengths of 100 as in the previous case, but the ratio of K_B to K_R is 1 (i.e., Blue weapons are as effective as the Red ones), the numerical solutions to these equations would be as shown in figure 5.

Figure 5. Lanchester Mixed Law chart



As shown in figure 5, the Blue force is at a significant disadvantage when being ambushed by the Red force. Although their weapons are equally effective against each other, the Red force inflicts far greater damage because of the target-rich environment presented by the Blue force under ambush. The Blue force is caught by surprise and can only return fire to the “general area occupied by the Red force” while the Red force has a full view of the Blue force and can engage with lethal “aimed fire” [19].

Next, we consider another functional form of the Lanchester-type equation, the Logarithmic Law.

The Logarithmic Law

If we assume that $r = v = 0$, $s = u = 1$, and $\alpha = \beta = 0$, we get the Logarithmic Law functional form of Lanchester-type equations as shown below:

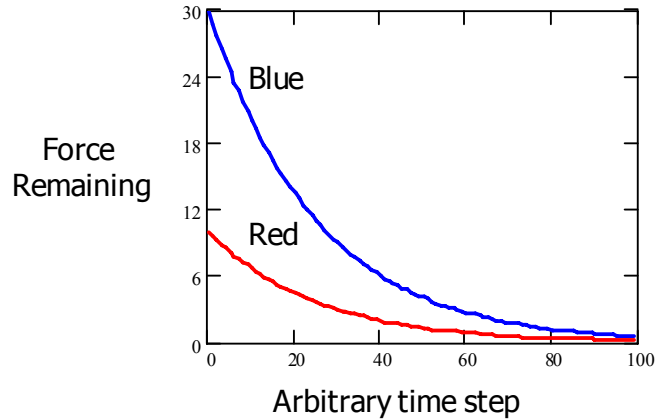
$$\frac{dR}{dt} = -K_B \cdot R \qquad \frac{dB}{dt} = -K_R \cdot B$$

It is obvious that the Logarithmic Law takes its name from the state equation shown below:

$$K_R \cdot \ln\left(\frac{R_0}{R}\right) = K_B \cdot \ln\left(\frac{B_0}{B}\right)$$

If we assume that the Red force starts with a strength of 30 and the Blue force starts with 10, and we assume the ratio of K_B to K_R is 1, we can plot the numerical solutions of the force strength over time as shown in figure 6.

Figure 6. Lanchester Logarithmic Law chart



This form of the Lanchester-type equations may be thought as a model for the early stages of a small-unit engagement (10-man Red squad vs. 30-man Blue platoon). During this early stage, both units were taken by surprise and vulnerable to enemy fire before they found effective cover. It is a situation in which the larger force presents more targets to enemy fire. As force vulnerability becomes the dominant factor in causing losses, a combat unit becomes more inefficient with increasing size [19]. As shown in figure 6, the larger Blue force suffers heavier casualties.

The Morse and Kimball case

In 1951, Morse and Kimball published their examinations on a special case of the Lanchester-type equations [20]. Based on the general form shown earlier, their special case assumes $r = v = 1$, $s = u = 0$, and positive values for N_R , N_B , α , and β . We get the following set of differential equations:

$$\frac{dR}{dt} = N_R - K_B \cdot B - \alpha \cdot R \quad \frac{dB}{dt} = N_B - K_R \cdot R - \beta \cdot B$$

The equations are very similar to those of the Square Law example. However, the additional terms allow the modeling of much more complex battles. For example, the terms N_R and N_B allow a model to include the element of replacement/reserve troops in a battle. Furthermore, factors α and β can be thought of as the impact of indirect fire by enemy forces or operational loss due to illness and non-combat related injuries.

Summary

As shown above, various forms of the Lanchester-type equations can be used to model a rich set of combat scenarios, and are still widely used in many of today's computerized combat models. However, Lanchester-type equations are based on many simplifying assumptions, which are discussed below.

Key assumptions

Lanchester equations represent highly abstract versions of combat, which are based on many simplifying assumptions, a few of which are shown below:

- Two homogeneous forces are engaged in a fire fight. Members of each force have the:
 - same weapon systems
 - same fighting capabilities
 - same vulnerabilities.
- Each unit on either side is within weapon range of all units on the other side.
- Effects of successive rounds in the target areas are independent.
- Both forces have constant lethalties (i.e., K_B and K_R are constants).
- Time spent on searching for surviving enemy units is independent of enemy force level.
- Fire is uniformly distributed over surviving enemy units.

Many of these assumptions are probably appropriate for the type of static trench warfare in the European theater during World War I. However, these assumptions are inadequate to address many of today's scenarios. For example, the assumption that a force is homogeneous is contrary to real-world observations of diverse individual behaviors in a battle. Therefore, a new approach is needed to analyze these problems.

Agent-based models offer a new approach

In contrast to Lanchester-type equations, multiagent-based models like ISAAC and EINSTEIN are based on the mathematical framework generally classified as *cellular automata* (CA), which can be defined as “a class of spatially and temporally discrete, deterministic mathematical systems characterized by local interaction and an inherently parallel form of evolution” [21].

First introduced by the mathematician von Neumann in the early 1950s to act as simple models of biological self-reproduction, CA are prototypical models for complex systems and processes consisting of a large number of identical, simple, locally interacting components. The study of these systems has generated great interest over the years because of their ability to generate a rich spectrum of very complex patterns of behavior out of sets of relatively simple underlying rules. Moreover, they appear to capture many essential features of complex self-organizing cooperative behavior observed in real systems.

ISAAC and EINSTEIN have been followed by many other cellular automata-based combat models. Because of their origin in complexity theory, EINSTEIN and other multiagent-based models can offer a fundamentally different approach to study a combat scenario as what it is, i.e., a complex adaptive system. In comparison, many conventional models can only pre-program a finite number of behaviors in a combat scenario. We explore EINSTEIN in more detail next.

This page intentionally left blank.

“EINSTEIN” goes to war

In this section, we first review results from recent CNA ground combat studies using EINSTEIN. The studies show how EINSTEIN can be used effectively to model small-unit combat scenarios.⁷

Review of CNA ground combat studies

Ground Combat Studies (GCS) refers to a series of CNA-initiated projects looking at ground combat for the USMC at the small unit (squad and fire team) level. As part of the project, a model for the simulation of ground combat was chosen to study the factors that most influence the composition of squads and fire teams for the Marine Corps as well as factors that influence outcomes of specific combat scenarios. After evaluating a number of models, the multi-agent-based model EINSTEIN was selected to simulate the ground combat scenarios typical of what Marines face on the battlefield.

EINSTEIN is an “agent-based” model where agents, which represent individual soldiers, are programmed with “personalities,” which influence them to act or react in certain ways to a number of perceived situations. For example, the agents in many of the studies in this project were five times more likely to advance towards enemy agents than towards friendly agents when healthy, while injured agents tended to move towards friendly agents and avoid enemies.

By setting up relatively simple “rules,” which define the personalities of the individual agents, and allowing the agents to interact with each other over time, information about the larger scale behavior of groups of agents can be collected and analyzed. In addition to setting

7. EINSTEIN is available for download from CNA’s website at <http://www.cna.org/isaac> [34]. In addition to the software program, the website also includes detailed user’s guide and an extensive collection of reference resources.

the personalities of the squad members, other variables such as weapon types and distribution, communications ability, and terrain effects can be set up in the program, and varied across multiple simulations. Results of the battles between opposing forces can be analyzed (generally in terms of attrition curves or a probability distribution of agents remaining at the end of a simulation), and trends in these data can be determined as a function of changing variables or parameters over the course of a number of simulations.

It should be noted that results for scenarios or battles are given over an average of many simulations, or runs, of the battle. Due to the stochastic nature of the simulations, the scenarios may be subject to run-to-run variability where either force may win (or more generally, one specific outcome occurs out of many possible outcomes). Attrition results are usually presented from an analysis of the overall averaged results, but probability distribution functions derived from the data can give some information about the variability of the results from run to run. While overall trends in the data are necessarily derived from the averages of the results, interesting insights into the dynamics of a scenario can often be obtained from observing individual runs in which the outcomes are highly variable.

Modeling a wide range of scenarios

One of the first goals in using EINSTEIN to model USMC small unit engagements was to simulate the wide range of different combat scenarios that Marines may encounter on the battlefield. A study in February 2005 [22] attempted to model seven different small unit scenarios, including (1) patrolling in formation, (2) reconnaissance, (3) infiltration of an enemy position, (4) security operations, (5) performing an ambush or reacting to an ambush (anti-ambush), (6) call for fire, and (7) fire and movement or bounding overwatch maneuver. These scenarios were selected from information derived from the *Guidebook for Marines* [23].

In this study and in subsequent studies, a “baseline” scenario was set up for simulation where all agent personalities, squad number and make up, enemy personality and force make up, weapon types and distributions, sensor and communication ranges, terrain effects, etc.,

were defined for the specific scenario. Excursions from the baseline scenarios were then performed, and the results from the excursions were compared to results from the baseline scenarios. Excursions generally aimed to change a key parameter or variable thought to be important to the outcome of a given scenario (e.g., having communications turned on or off). These seven scenarios were used as a proof-of-principle to show that EINSTEIN could simulate combat simulations and provide results that were “realistic” and could reproduce conclusions and trends that coincide with the experience of Marines based on exercises and real combat situations. From the seven scenarios listed above, patrolling, reconnaissance, security, ambush/anti-ambush, and call for fire were all simulated successfully in EINSTEIN.

In one of the examples, the “Call for Fire” scenario, a Blue squad was created that was comprised of four agents and a Red squad was created that consisted of 13 agents. The Blue agent personalities were defined so that the agents were more likely to advance towards enemy agents than friendly agents. However, if the Blue agents came under fire from the Red squad, a “meta-personality” was activated in the Blue agents and they backed away from the enemy.⁸ Once the agents backed away from the enemy, they were able to communicate with a defined “super-agent,” which represented a strike plane that would provide tactical air support. The agent representing the plane was able to move towards the enemy at a rate four times what the normal agents could move, and possessed a high power area weapon representing a 2,000-lb Joint Direct Attack Munition (JDAM) bomb. The plane was also defined as being invulnerable to attack by the enemy (although the authors point out that they could simulate the plane's vulnerability to enemy air defenses by reducing the armor rating of the plane). Thus, when the Blue squad made contact with the Red squad, the Blue agents called in an air strike and the plane moved to attack the Red agents. The Blue agents, then, were able to advance to attack and finish off the injured Red agents. In this scenario, an agent was used to represent a vehicle in the simulation (although, the authors admit, in a crude way), which shows the ability of users to

8. For more description of assigning personality traits and meta-personalities, see the following section titled “Basic Setup for Using EINSTEIN”.

adapt EINSTEIN to the specific details of the scenario they want to simulate. Indeed, the wide variety of scenarios modeled in these seven scenarios demonstrate the versatility of the model.

Even though EINSTEIN showed the ability to model a number of different scenarios for small unit combat, there were scenarios that EINSTEIN was unable to realistically model in this study. Both the enemy infiltration scenario and the fire and movement scenario were unable to simulate the behaviors of fire teams executing the given tasks. The most likely reason that these scenarios could not be modeled was that the basic personalities and parameters assigned to the agents in the squads in this study did not produce more complex coordination and problem-solving behaviors needed to perform these two scenarios. For example, in the fire and movement scenario, the Blue squad was split into two teams against a single Red squad. One Blue team is supposed to engage and fire at the Red squad while the second Blue team loops around the Red squad and attacks their flank. However, during the simulation, the teams did not act in a way consistent with how real teams would act. This is because, in EINSTEIN, the concept of fire suppression is not easily modeled, and the Red squad under attack does not “hunker down” when fired upon. Furthermore, the rules used in these scenarios have the agents shoot at the enemy whenever they can see the enemy and whenever the enemy is in weapons range. The fire and movement and the enemy infiltration scenarios require the “restraint” of the units to not fire at enemies when they see them, but to perform movement to either avoid enemies (for infiltration) or to get into better position first before firing at the enemy (for fire and movement). More complicated personality profiles could probably be created to allow the successful modeling of these scenarios, but this was beyond the scope of the study.

Defensive combat modeled in EINSTEIN

CNA analysts tried to model defensive combat scenarios using the terrain feature in EINSTEIN. Defensive forces were set up behind terrain features while attacking forces attempted to overrun the position to capture the defender's flag.

A simple example of this scenario had a force stationed behind a simple berm, which reduced their visibility to attackers [24]. When both forces were the same size and possessed the same weapons, the defenders generally lost fewer troops than the attackers and were defeated in only a small number of the runs. By varying the number of attacking agents, the model demonstrated that an attacker-to-defender ratio of approximately 3:1 was needed for the attackers to routinely overrun and defeat the defenders. This result is in agreement with the general rule of thumb⁹ for attacking forces from real life combat experience, and showed the ability of EINSTEIN to model this scenario realistically and derive the same conclusions as experienced Marines.

This result led to recommendations by Headquarters Marine Corps, Programs & Resources division for further study of a defensive combat scenario, in which a Blue force defends terrain in a rocky environment against an attacking enemy Red force [27]. The baseline scenario defined a larger attacking Red force that included squads of 60 troops with “standard weapons” (rifles, grenade launchers, etc.) and 60 irregular troops with rifles and rocket propelled grenades (the irregular troops also had a “tremor factor” that reduced the accuracy of their weapons). The attacking force had tanks (125-mm cannon) and infantry vehicles (30-mm cannon). The Blue force consisted of 45 troops in the defense that possessed four anti-tank missiles (TOWs¹⁰ and Javelins) as well as two .50 caliber machine guns. The rocky terrain around the Blue force reduced their visibility to the attackers by roughly 60 percent. Excursions from the baseline scenario included varying the weapon types and number, varying the number of attackers, and varying the “intelligence” of the defenders (with emphasis on how and when weapons were used).

In the baseline scenario, the attackers overran the defenders 70 percent of the time. However, by observing results from individual runs,

9. Origin of the “3:1” rule in ground combat is not clear [25]. Some have attributed it to a letter from President Lincoln to Union Army Major General Halleck during the Civil War [26].

10. TOW stands for Tube-launched, Optically tracked, Wire-guided.

the authors of this study noted that the defenders were able to hold off the attackers when the small number of Blue agents with the .50 caliber machine guns avoided being killed early in the battle (a stochastic event). Thus, the excursion scenarios were designed to examine the effect of adding more machine guns as well as changing the personalities of the Blue agents to defend the machine gun wielding agents at the beginning of the battle until the tanks and infantry vehicles could be neutralized.

The results of these excursion scenarios demonstrated the importance of protecting high value assets, that proper targeting of enemies is important, and that proper training and tactics are more important to overall success than simply adding more firepower. Much like the earlier defensive scenario, these results coincide with the expectations and experiences of Marines in real life combat scenarios.

Combat in open terrain and urban warfare

Based on the demonstrated abilities of EINSTEIN to simulate the wide range of scenarios discussed above, the Commandant of the Marine Corps requested that the GCS model ground combat in an open environment and in an urban setting [28]. In the baseline scenario for the open terrain environment, the Blue and Red forces were of equal size and had similar personalities and weapons. The Blue force was designed with 60 agents broken up into 5 squads, while the Red force consisted of 60 agents arranged either as one large group, 5 squads of 12 agents, or fully dispersed throughout the battlefield. For each force, weapons consisted of rifles, machine guns, and grenade launchers (distributed in a 2:1:1 ratio, respectively). The Blue squads had the ability to communicate with each other, while the Red troops had no communications capability.

The main objective was to study the effects of initial enemy dispersion on the performance of the Blue squads in the battles. Excursions from the baseline scenario included changes in weapon types, sensor range, visibility, and communications ability. For the simple rules describing the agent personalities in these simulations, it was apparent that anything facilitating location of the opposing force led to

more combat and higher attrition rates, while limiting the ability to find the opposing forces led to less combat and lower attrition. For example, dispersing Red agents homogeneously over the battlefield led to more Red attrition than dividing the Red into five squads because it was easier for the Blue squads to find the Red agents when they were completely dispersed. Other excursions demonstrated the benefit of technology, where agents with “night vision” could easily dominate an enemy without this technology. In these scenarios, sensor ranges were very short for Red agents (simulating their limited ability to see in the dark) while Blue agents had much larger sensor ranges simulating the use of night vision equipment. Visibility and weapons capability were determined to be important factors in the results of these scenarios.

An interesting and counter-intuitive result from these simulations was that communications ability for the Blue agents actually led to higher Blue attrition than not having any communications. It is generally expected that forces will do better with the ability to communicate with each other, but, in this case, communications only made it easier for Blue squads to locate the Red agents and engage in combat, and communications weren't used to coordinate action between the different squads, where the benefit of communications lies. Such coordinated action is an example of a higher order behavior that is not observed with the simple rules programmed into the agents in these scenarios.

Following on the results of combat in open terrain, the study next looked at the effects of five Blue squads against differently distributed Red forces in an urban environment. In these scenarios, buildings were simulated as obstacles arrayed regularly in a grid, through which agents could neither move nor see through. The buildings were set up with a 5-by-5 size with 2 battlefield units separating each building. Given the size ratios of building size to battlefield size, these buildings would be more accurately described as city blocks and not individual buildings, although the authors were not convinced that finer resolution was needed to capture the essential trends in the data. For these simulations, the baseline scenario was the same as described for the open terrain scenario (except for the inclusion of the buildings). Excursions from the baseline included varying street size, building

size, communications ability, visibility (fighting at night), weapons capability, and using patrols of Blue agents around the battle space. The results of these simulations were similar to the open terrain scenarios in that combat and attrition rates are ultimately determined by the ability of the forces to find each other. Overall, compared to the open terrain scenario, there was much less combat and attrition because the buildings limited the sensor and weapon ranges, and made it harder to find and fire upon opposing forces. These results and observations agree with the conclusions from the Hunter Warrior and Urban Warrior experiments conducted by the Marine Corps Warfighting Lab (MCWL) [28].

Although the conclusions reached by the urban warfare simulations agreed with similar conclusions from MCWL, there were some aspects of the model which could not capture real soldier behaviors. One problem had to do with the agents having no “memory.” For example, if a Blue agent was chasing a Red agent, the Blue agent would stop his chase if the Red agent went around a building corner and out of the Blue agent's sensor range. Also, if a Blue agent was killed going around a corner by a Red agent, other Blue agents in the same area would continue going around the corner instead of reacting to the sight of another agent being killed just a few moments earlier.

Discussion of the use of EINSTEIN for the GCS

Overall, the EINSTEIN simulations were able to recreate important trends in ground combat and identify important factors affecting the outcomes of certain battle scenarios. The determination of the 3:1 attacker to defender ratio needed to take a defensive position, the identification of important tactics such as protecting high value assets, and the conclusion that a drop in weapon and sensor range will greatly affect urban combat are important proofs that the EINSTEIN model can simulate important aspects of real world combat.

Thus far, the goal of the GCS has been recreating known trends. A more far reaching goal would be to use EINSTEIN to predict trends or reach novel conclusions from combat scenarios where little, if any,

empirical data exist. However, one must be careful when trying to produce trends without external empirical support. This is ultimately very difficult because it requires the user of the model to be able to differentiate between a real trend and something that is an artifact of the model (and hence, not a trend in the real world). For example, the required 3:1 ratio of attackers to defenders is an EINSTEIN-world result which maps onto the real world as well. However, the use of communications actually hindering a force as opposed to having no communications is an EINSTEIN-world result that is known not to be true in the real world for reasons discussed above. A key step in confirming that this model can be used more predictively will be the ability to model more complex behavior within and between the agents, and thus making the EINSTEIN world more similar to the real world. However, the problem of deciding the “truthfulness” of a prediction from EINSTEIN will still be a fundamental problem.

A related issue seems to be the extent to which the agents can be thought of as “people” and the extent to which the user needs to remember that the agents are just “robots” that are programmed to react in a certain way to certain stimuli, keeping in mind that these stimuli are stochastic in nature. The simple defining of the personalities of agents can make them rush toward an enemy or retreat from an enemy, but the rules in EINSTEIN needed to produce more complex behavior such as coordinated movement between squads to attack an entrenched enemy may be unknown and difficult to discern with the current methodology of setting a base scenario and taking relatively minor excursions from the scenario by changing a few key variables (although a skilled user of EINSTEIN may be able to produce these behaviors with intelligent choices for the personality and trigger state definitions).¹¹

Interestingly, there is a component to EINSTEIN that allows a user to iteratively optimize a force to achieve a well-defined goal or mission. The genetic algorithm (GA), in principle, allows a user to define a goal or set of goals (such as maximize Red casualties, or minimize Blue casualties, etc.) and let the algorithm search a given parameter

11. The next version of EINSTEIN is designed with a new command and control module to allow modeling of more complex scenarios.

space to develop the characteristics of the squad or squads that are “optimal” in achieving the goal or mission. For example, in trying to create an infiltration scenario, the goal would be to create a squad or squads that can minimize the time required to get to a point (x,y) on the battle space while also minimizing the number of Blue casualties. The algorithm would then “synthesize” or “breed” a squad or squads that could best achieve these goals (i.e., these squads have certain values or ranges of values defining their personalities that allow them to perform some stated goal or goals with a level of proficiency). Relevant questions to ask once an optimal force is created for these simulations may include:

- How do the squads achieve their mission?
- What are the personality profiles of the squad members and how would one qualitatively describe them?
- How sensitive are the results to small changes in some of the perceived key variables in the model?

Of course, there may be many limitations to this approach (not even considering computer time needed and the ability to accurately define a set of goals or missions for the algorithm to work towards). It may be very difficult to interpret the results of such an optimization. If the combination of parameters and personality traits for the optimized force is sufficiently complex as to be not understandable by a simple qualitative (or quantitative) analysis, can the computational and analysis time be justified? Additionally, if “sensitivity analyses” should be run on a number of variables, how can one identify what variables should be “key variables”? How can one reconcile a sensitive variable that does not make logical sense to an analyst or a variable that can be altered in EINSTEIN, but does not have an easily identifiable real world analogue (for example, if soldiers only with a certain value or set of values for “advancing towards enemy” can complete a given mission)? These results may not produce meaningful conclusions for the analyst and, most importantly, for the USMC.

Updates to the program over time may help in improving the realistic nature of the simulations. In addition, they may make it easier and perhaps simpler to implement some of the earlier combat scenarios.

Changes in the program as described recently [31] include a memory function for the agents, a greater range of health states, a dynamic targeting logic for the agents, and a more dynamic refinement of meta-states (trigger states) that could result in more complex behavior and actions for an individual agent or groups of agents. For example, in the most recent article about EINSTEIN from Andrew Ilachinski, an example of a trigger state called “Hunkered Down” is given where presumably these agents would take up defensive positions when under enemy fire. This type of behavior was not easily induced in previous versions of EINSTEIN and led to some troubles in modeling certain combat scenarios. A variety of other trigger states may be defined that can model “real” behaviors of soldiers and squads on a battlefield. Furthermore, the addition of command and control features in future updates will significantly enhance the ability of the user to model more “realistic” combat scenarios.

Comments on VV&A

As shown earlier, models play important roles in many areas of the DoD community. In order for models to be trustworthy, DoD requires that they go through a verification, validation and accreditation (VV&A) process [35, 36, 37]. Definitions of these terms are shown below.

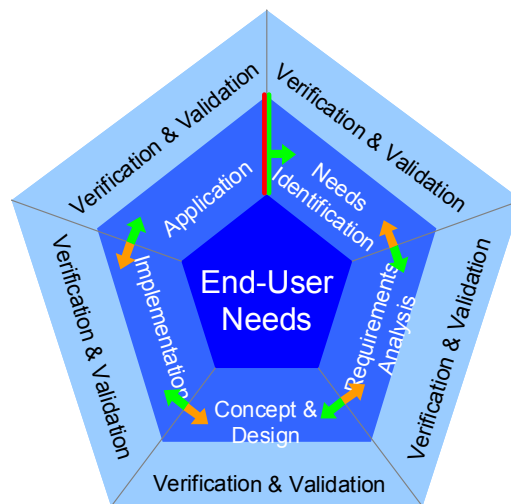
“Verification is the process of determining that an M&S accurately represents the developer’s conceptual description and specifications. Verification evaluates the extent to which the M&S have been developed using sound and established software-engineering techniques.

Validation is the process of determining the extent to that an M&S is an accurate representation of the real world from the perspective of the intended use of the M&S. Validation methods include expert consensus, comparison with historical results, comparison with test data, peer review, and independent review.

Accreditation is the official determination that a model, simulation, or federation of M&S is acceptable for use for a specific purpose” [37].

As the definitions suggest, the VV&A process is to be implemented throughout the developmental steps of a model which is shown in figure 7. As shown in the figure, the end-user needs should always drive the modeling processing.

Figure 7. Modeling steps and VV&A



The most challenging part of the VV&A process is probably the middle “V”, validation. However, as Ilachinski pointed out [31], since EINSTEIN is designed as “an exploratory model, validation is less of an issue than it might be for more ostensibly realistic models. As long as EINSTEIN’s outcomes are intuitive and its inputs are easy to generalize (or translate) for use by more realistic models, EINSTEIN will remain useful for many different kinds of exploratory analyses.”

Summary

Overall, EINSTEIN has shown its efficacy and versatility as a model for ground combat by small units. A wide range of different small unit engagement scenarios has been simulated in EINSTEIN and important trends can be discerned that correlate well with empirical data from exercises and historical accounts of battles. For example, the 3:1 attacker to defender rule of thumb, the importance of better tactics

over better firepower, and the conclusions about limitations in urban combat are important results and show EINSTEIn can accurately model certain aspects of ground combat.

Although EINSTEIn, as with any model, can only approximate a real battlefield, information that has some mapping onto real combat situations can be easily extracted by analysts. EINSTEIn is much more than a quick visualization tool, the program is designed as a capable data collection tool. Our review of the recent CNA studies and the tutorial in the next section show only a few of the rich features available in EINSTEIn.¹² Some of the key features that we did not get a chance to explore include “intelligent route-finding, scriptable waypoints, and programmer access to all of EINSTEIn’s 300+ source code level functions using EINSTEIn’s recent port to the Python scripting language¹³” [29].

Basic setup for using EINSTEIn

In this section, we will discuss the basic setup of a battle, or run, in EINSTEIn, show how to assign “personalities” to agents, and highlight some interesting sample scenarios to demonstrate the diversity of the model for company/battalion-scale engagements involving many agents. For a much more complete discussion of EINSTEIn and how to use the program, see the publications from Andrew Ilachinski [31, 32, 33].

-
12. Our limited scope also did not permit us to review EINSTEIn’s applications outside of ground combat. In fact, EINSTEIn and other agent-based models have shown wide applicabilities covering a variety of military and non-military problems. For example, CNA has led “a series of workshops at the Naval Postgraduate School applying EINSTEIn to a maritime environment” (e.g., swarm attack tactics by small boats on a high value unit) [29, 30].
 13. For example, EINSTEIn’s Python extension allows a batch of multiple simulations to be run simultaneously on a single computer. This dramatically increasing its data collection capability [29].

Setting up the battlefield and forces

The battlefield size in EINSTEIN can be set using the *Edit > Battlefield Dimensions* tab in the toolbar menu. The dimensions of the battlespace can be entered in the appropriate fields for the length and width (these distances are given in arbitrary units). Locations on the battlefield such as the flag and goal positions are designated with an x-y coordinate within the battlespace.

The user can create a single or multiple squads of agents for either of the two sides (Red or Blue) under the *Edit > Force Parameters* tab in the toolbar menu. The user can specify both the number of squads and the number of agents assigned to each squad. Additional force-wide parameters can be assigned under the *Edit > Force-wide Parameters* tab in the toolbar menu. Here, the coordinates for the flag of each force can be assigned and other parameters such as the logic for moving around terrain elements and the communications ability can be assigned. The terrain movement logic describes the way agents move through the battlespace when terrain elements are introduced, and the communications ability allows agents to communicate information to other agents over distances longer than their normal sensor ranges. The examples in this section do not enable the terrain movement options or communications features, but other CNA publications in the GCS discuss the use of these features within combat scenarios [22, 24, 27, 28].

Assigning squad personalities

The initial battlefield and force parameter assignments described above create the two forces and the basic battle area for these forces. At this point, starting the simulation will result in the agents randomly moving around within a small area near the starting point (this is actually the default scenario that loads when opening EINSTEIN for the first time). In order to get the agents and forces to do something “interesting,” personalities must be assigned to the agents.

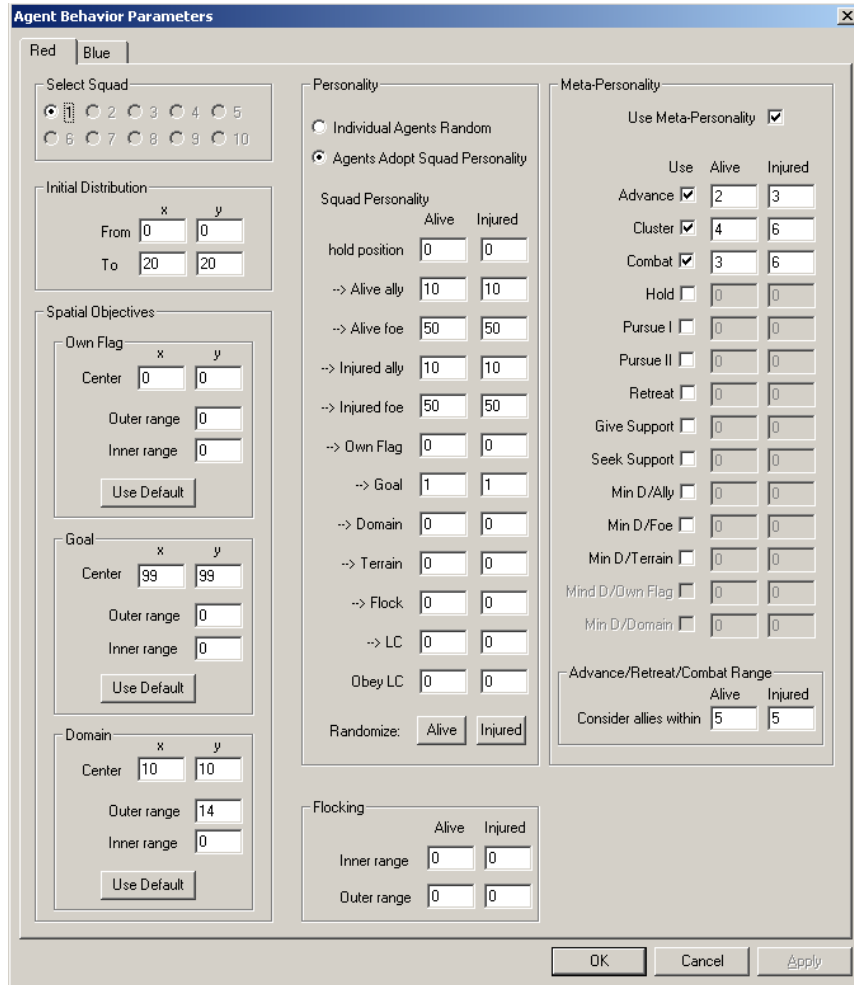
Agent personalities are developed through “weighting factors” for certain actions an agent can take on a given time step during the simulation. The action chosen by an agent is based on a “local penalty function,” which the agent uses with the weighting factors to deter-

mine the “best” action to take during a given time step. A much more thorough discussion of the general decision logic of the agents is given in another document [33]. Basically, the relative weights given to possible actions influence the agents to move in a certain direction or to fire at a certain target. For example, a couple of basic personality traits are the “desire” to move towards an enemy agent or to move towards a friendly agent. Depending on the relative values assigned to these two traits, the agent may chose to move to a location on the field closer or farther away from enemy agents (or friendly agents). There exist a number of parameters in an agent's personality that can be assigned weighted values, but for the purposes of this report, we will only describe a few basic parameters needed to get an unfamiliar user started using the program.

Under the *Edit > Agent Behaviors > Spatial Orientation/Movement* tab on the toolbar menu, a table listing the parameters for the agent personality weights and the spatial distribution parameters for the squads is displayed (see figure 8).

The flag position can be set in this window (as well as in the Force-wide parameters window), and the coordinates for the goal of each squad can also be set. Notice the goals do not have to be the same locations as the opposing flags. The initial location and distribution of both forces can be assigned as well, and the forces can be initially positioned either near the flag/goal or away from these locations. The forces can be either narrowly or widely distributed at the beginning of the simulation.

Figure 8. Agent behavior parameters window



The middle portion of the window in figure 8 displays the “traits” for the agents' personalities with input windows for the trait weighting factors. Notice that the agents in different squads can have different personalities, but, currently, all members of the same squad must have the same user-defined personality. There are 12 personality traits listed in this window, but in the sample scenarios presented in this document, we will only assign values to 6 of these traits (with the other traits having weights of zero). There are 2 basic “states” which describe the health of the agent over the course of the simulation. Agents are either designated as “alive” or “injured” and the personal-

ity traits can be assigned specifically for the different health states. An agent can also be dead, of course, and these agents disappear from the battlefield. The personality parameters used in the examples given below are:

- Alive ally: weight for agent to move towards an alive friendly agent
- Alive foe: weight for an agent to move towards an alive enemy agent
- Injured ally: weight for an agent to move towards an injured friendly agent
- Injured foe: weight for an agent to move towards an injured enemy agent
- Own flag: weight for an agent to move towards own force's flag
- Goal: weight for an agent to move towards the force's goal.

In addition to the parameters listed above, the agents can also be assigned “meta-personality” parameters. An agent meta-personality is an adjustment to the default personality based on the specific circumstances an agent perceives in the local environment. The user can specify some threshold constraints that have to be satisfied prior to the activation of the meta-personality. For example, for the combat meta-personality, an agent will assess the difference in the number of friendly agents and enemy agents in a given range (which can also be assigned) and attack the enemy if the difference is greater than the number assigned for this parameter. This number can be positive or negative so that agents may only attack an enemy with numerical superiority (positive threshold) or they will attack the enemy even if they are outnumbered (negative threshold). The meta-personality parameters used in the sample scenarios are:

- Advance: An agent will advance toward the goal if a minimum number of friendly agents is near.
- Cluster: An agent will stop moving towards friendly agents if the number of friendly agents exceeds a given threshold.

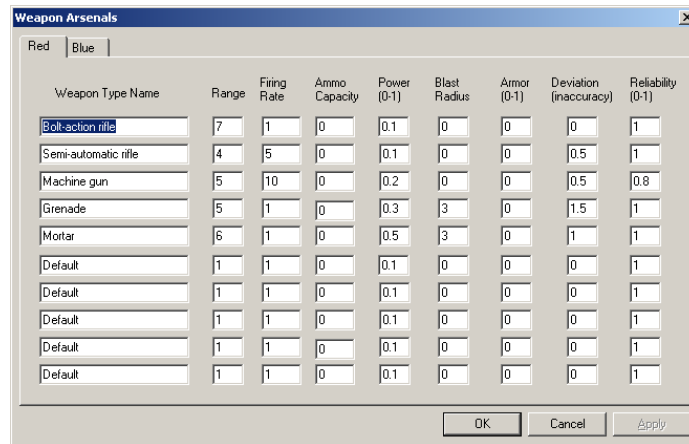
- **Combat:** An agent will move towards an enemy if the difference between friendly agents in the area and enemy agents in the area exceeds a given threshold.

Further discussion of these and other personality parameters and meta-personalities is given in another report [33].

Weapon arsenals and assignments

Up to ten different weapon types, five of which are predefined, can be used by agents in an EINSTEIN model. The weapon arsenal window can be seen by selecting the *Edit > Weapon Arsenals* tab in the toolbar menu (see figure 9).

Figure 9. The default weapons arsenal in EINSTEIN



As shown in figure 9, weapons in EINSTEIN are characterized by the range (in battlefield units), firing rate (rounds per time step), ammo capacity, power, blast radius (for area weapons), armor (amount of protection they provide), deviation or inaccuracy, and the reliability. Any of the values can be changed for the previously defined weapons, and new weapon types can be created by changing the values (and the name) for any of the Default weapons shown in the bottom five rows in figure 9. For example, in some of the scenarios described below, we define a “shock weapon” that has short range of one unit and power of 0.05 to enable close range combat.

The user can assign these weapons in any distribution to the agents in a squad by choosing the *Edit > Weapon Assignments* tab in the toolbar menu. An agent can be assigned at most one weapon so that the total weapons assigned cannot be more than the total number of agents in a squad, although the user may assign fewer weapons than squad members. The assigned weapons are distributed randomly to the agents in the squad.

Movement and sensor ranges

The final parameters used to set up the sample scenarios for this section are the movement and sensor ranges. These ranges can be assigned by choosing the *Edit > Agent Capabilities* tab in the toolbar menu. Movement range is the number of battlefield units an agent can move in a time step, while the sensor range is the number of battlefield units an agent can “see” prior to making a decision to move in a direction or to fire on an enemy. These ranges can greatly affect the dynamics of the simulations as discussed below. Other parameters in the window are not used in this report, but are discussed in more detail elsewhere [31, 32, 33].

Terrain elements

Terrain elements, which restrict movement and line of sight (LOS), can be introduced into the battlefield environment. Terrain can be drawn onto the battlefield by clicking the right mouse button and selecting “Draw Terrain” from the menu. A number of options come up with different movement and LOS restrictions. More information about these terrain types can be found under the *Edit > Terrain Types* tab in the toolbar menu. The numerical values for “passability” and visibility are given for the default terrain types, and these values are the fraction by which the movement and sensor ranges are reduced for agents near the terrain element. User-defined terrain elements can also be created in this window by selecting the “New” tab and inputting the values for passability and visibility and naming the terrain element.

Examples of battle simulations using EINSTEIN

For the sample scenarios presented in this section, we demonstrate company/battalion-scale troop behavior similar to the original intended use of EINSTEIN (and ISAAC) for exploring emergent behavior in complex systems [33]. This is different than the previous use of EINSTEIN for the GCS, which studied small unit combat at the squad and fire team level.

In the following scenarios, we use a single squad consisting of 300 agents for both forces, and show that the assignment of simple personality values can result in the observation of complex behaviors of groups of agents. Furthermore, varying a small number of these parameters can lead to different overall behaviors of the forces, which demonstrates the diversity and flexibility of EINSTEIN.

Example 1: Linear battle fronts

The first example we discuss is similar to a case study in an earlier CNA publication [33]. The key agent parameters for this example are listed in table 1.

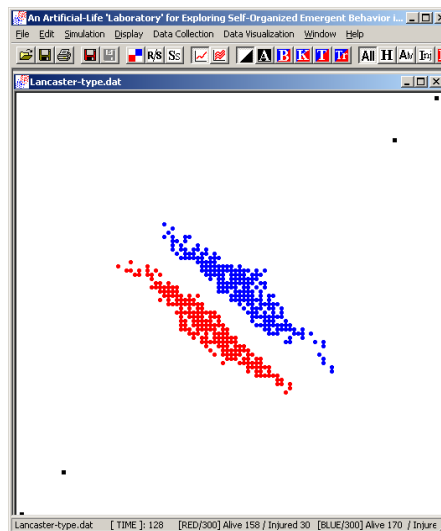
Table 1. Example 1: agent parameters

Agent parameter	Alive (Healthy)	Injured
Alive ally	10	10
Alive Foe	50	50
Injured Ally	10	10
Injured Foe	50	50
Own Flag	0	0
Goal	1	1
Advance	2	3
Cluster	4	6
Combat	3	6
Movement range	1	1
Sensor range	10	10
Weapon range	7	7

The Red and Blue forces are given identical personality weights and weapon assignments (the weapon range listed in table 1 is just the range for the default bolt action rifle). Both forces were situated at their own flag at the beginning of the run (Red in the bottom left corner, or (0, 0) on the battlefield grid and Blue in the upper right corner, or (99, 99) on the grid).

At the beginning of the run, the forces advance towards the center of the battlefield as they make their way to their respective goals (the opposing force's flag). As the forces move within sensor range of each other, they form two lines stretching across the battlefield and begin firing at each other. Figure 10 shows the configuration of the two forces approximately 130 time steps from the beginning of the run.

Figure 10. Example 1: screenshot



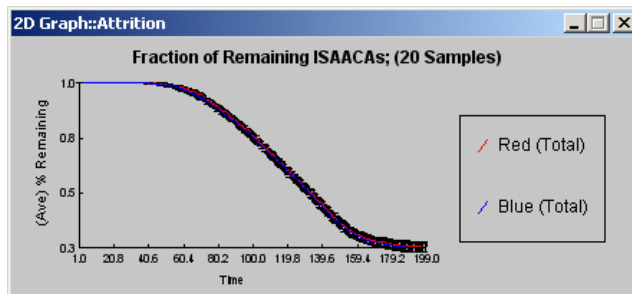
At this point in the simulation, both forces have lost ~ one-third of their forces. As the simulation continues, both sides take losses until the battle breaks and both forces rush to the opposite flag. The stochastic nature of the model precludes one from predicting which side will lose more troops during a single run in this scenario. However, by running a number of simulations and analyzing the average results, conclusions can be made about the average performance of each force.

Results for these runs are usually presented in terms of attrition over time, but a number of different measures can be used to evaluate a particular scenario. The different types of data that can be collected can be found by selecting the *Data Collection* tab on the toolbar menu. These data types are further described in other publications discussing EINSTEIN [31, 32, 33].

To collect data over multiple runs, the *Simulation > Run Mode > Multiple Time Series* tab on the toolbar menu can be selected. The user enters the number of runs and the number of time steps for each run. EINSTEIN then runs the simulations (without the graphical display of agents) once the user begins the simulation (from the *Simulation > Run/Stop* tab on the toolbar menu).

Once the simulations are complete, the data for attrition, for example, can be visualized graphically by selecting the *Data Visualization > 2-D graphs > Attrition* tab on the toolbar menu. Figure 11 shows attrition data averaged over 20 runs of this scenario.¹⁴ Data can also be saved and exported into spreadsheet programs like Excel for further analysis.

Figure 11. Example 1: Average attrition data^a



a. For 20 runs of the linear battle front example.

As shown in figure 11, attrition profiles of both forces are very similar. This should come as no surprise since their agent parameters are the same.

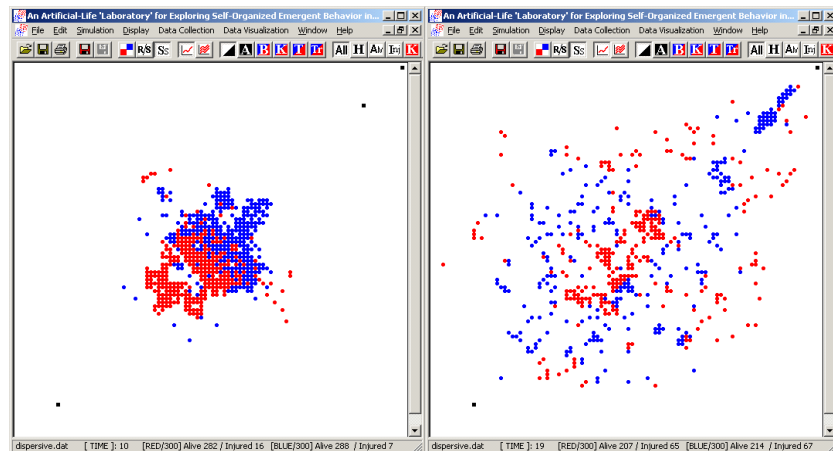
14. Each simulation lasts for 200 time steps.

Example 2: Dispersive battle

In the previous example, when the opposing forces approached each other, they stopped their movement toward the opposing flag and engaged each other in battle by forming firing lines in the configurations shown in figure 10. In that example, the agents' sensor and weapon ranges were much longer than the movement range. In this section, we present a second sample scenario where the sensor range and movement range are equal (values set to 5), while the other agent parameters are set to the same values as in the previous example (see table 1).

As the run begins, the forces move toward the opposing flags and approach each other as they did in the previous example. However, instead of stopping and forming firing lines, the forces run headlong into each other and a small degree of “mixing” or interpenetration of Red and Blue agents occurs where the two forces engage each other as shown in the left panel in figure 12. As the battle evolves, the agents in the two forces disperse into a random configuration of smaller fire-fights all over the battlefield as shown in the right panel of figure 12.

Figure 12. Example 2: screenshots^a



- a. The screenshot on the left shows the initial engagement of the two forces ($t=10$), while the screenshot on the right shows the agent configuration just nine time steps later ($t=19$).

Interestingly, the much different collective behaviors of the agents in the linear battle front example and in this dispersive battle example arise from a change in just two of the parameters (movement and sensor range) of the simulation. In fact, varying the ratio of sensor range to movement range causes a change in the collective agent behaviors from the short time scale, short distance order demonstrated in the dispersive battle example to a longer time scale and longer range spatial order as demonstrated in the linear battle lines example. The time scale and the length scale of the battles also are a function of the “killing rate” or loss of agents by attrition. Thus, much longer time scales and longer range order in the forces can be explored by reducing weapons range and power.

Example 3: Swarming Blue forces

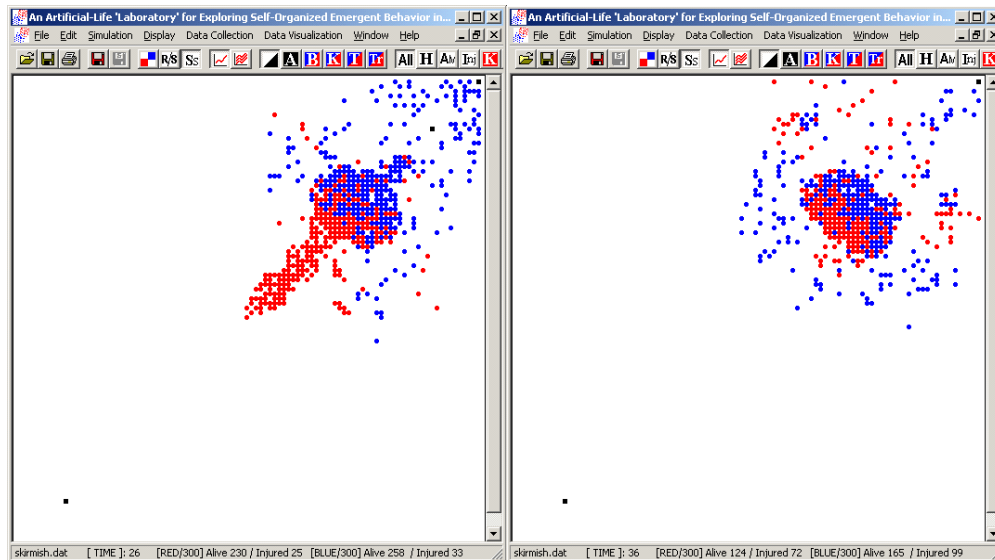
In this next sample scenario, the forces were made more “aggressive” by changing the combat meta-personality thresholds. The combat threshold for the forces is set to -99 and -3 for healthy and injured agents, respectively, so that agents will attack enemies even when outnumbered. Table 2 shows other agent parameters for this example.

Table 2. Example 3: agent parameters

Agent parameters	Red		Blue	
	Alive (Healthy)	Injured	Alive (Healthy)	Injured
Alive ally	10	10	10	10
Alive Foe	40	40	40	40
Injured Ally	10	10	10	10
Injured Foe	40	40	40	40
Own Flag	0	0	0	0
Goal	10	10	0	0
Advance	3	3	3	3
Cluster	8	3	8	3
Combat	-99	-3	-99	-3
Movement range	9	9	9	9
Sensor range	3	3	3	3
Weapon range	3 (50%) 1 (50%)	3 (50%) 1 (50%)	3 (50%) 1 (50%)	3 (50%) 1 (50%)

Notice that the Red force will advance to the Blue force's flag but the Blue force will “lie in wait” for the Red force since the Blue goal parameter is set to zero. In addition, half of the agents in each force were assigned user-designated “shock” weapons with a range of 1 and half of the agents in each force were assigned the bolt action rifle with the range reduced to 3. Screenshots of this example are shown in figure 13.

Figure 13. Example 3: screenshots^a

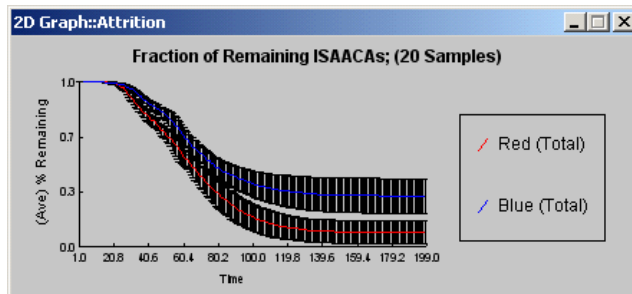


- a. The Red force in this example attacks the Blue flag. The Blue force first moves around the Red flanks as shown in the left panel and then envelops the Red force as seen in the right panel. The Blue force then moves in and destroys the Red force.

Once the simulation is started, the Red force advances toward the Blue flag. As the Red force moves into the sensor range of the Blue agents, the Blue force attacks by engaging the Red force head on while, simultaneously, some of the other Blue agents move around the Red flanks and eventually attack the Red force from the rear as well. The left panel of figure 13 shows the initial Blue attack including the agents moving around the Red flanks. A short time later in the simulation, the Blue forces have the Red force completely surrounded and are able to move in and destroy the Red force.

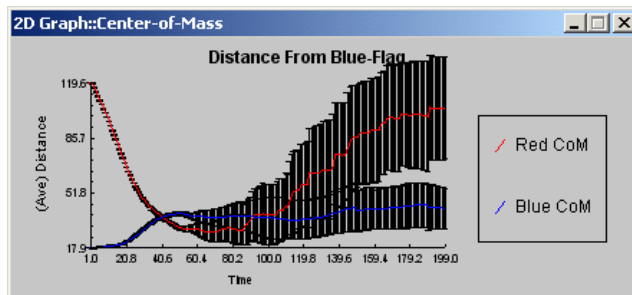
Looking at the attrition data for 20 runs of this scenario in figure 14, one can see that the Blue force dominates the Red force during this scenario (although the Blue force does incur significant losses).

Figure 14. Example 3: attrition data



Additionally, the Red force never makes it to the Blue flag after the initial approach, which can be seen by plotting the distance of Center of Mass (COM) of the Red forces from the Blue flag versus time as shown in figure 15.

Figure 15. Example 3: CoM^a distance from Blue flag

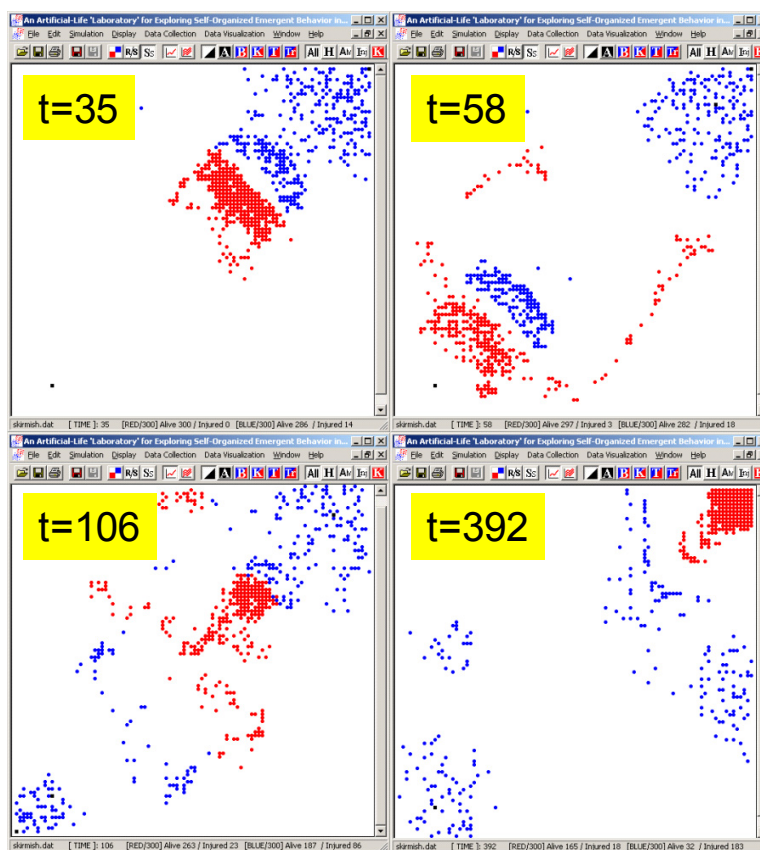


a. CoM stands for center of mass.

Example 4: Red feint retreat

The scenario depicted in example 3 resulted in the Blue force successfully repulsing the Red force attack on their flag. In the following example, we show how a slight change in the Red force's parameters results in a change of tactics by the force. In this example, the Red force will be made less aggressive by setting their combat meta-personality threshold to 20 and 5 for healthy and injured agents, respectively. For this example, the sensor range for both forces is set to 15 battlefield units and all other parameters are the same as in the previous example (see table 2). The evolution of this battle is shown in figure 16.

Figure 16. Example 4: screenshots time series^a



- a. Evolution of example scenario 4 as the Red force makes contact with the Blue force ($t = 35$) and appears to retreat as a smaller Blue force pursues ($t = 58$). The Red force maneuvers around the small Blue detachment and re-aggregates before the final attack on the Blue flag ($t = 106$). Now that the defending Blue force is smaller, the Red force takes the flag ($t = 392$).

As the scenario begins, the Red force moves towards the Blue force's flag similar to the Red force in the previous example. However, in this scenario the Red force halts before rushing into battle with the Blue force ($t = 35$, figure 16). At this point, the Red force appears to retreat back towards the Red flag, and a portion of the Blue force pursues the retreating Red force. However, as the Red force is moving back, Red agents begin to move around the flanks of the smaller Blue detachment ($t = 58$, figure 16). The Red force then outmaneuvers the smaller Blue attacking force by dividing into roughly two separate forces, moving around the Blue detachment and reforming in the middle of the battlefield for the attack on the Blue flag ($t = 106$, figure 16). The subsequent attack on the now smaller defending Blue force is successful as the Red force occupies the Blue flag ($t = 392$, figure 16). Figures 17 and 18 show the average attrition and average center of mass positions, respectively, over the course of 20 runs.

Figure 17. Example 4: attrition data

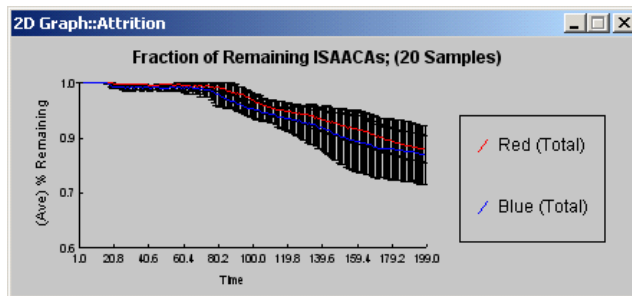
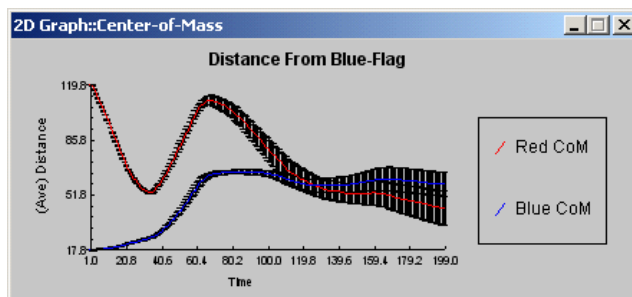


Figure 18. Example 4: center of mass data

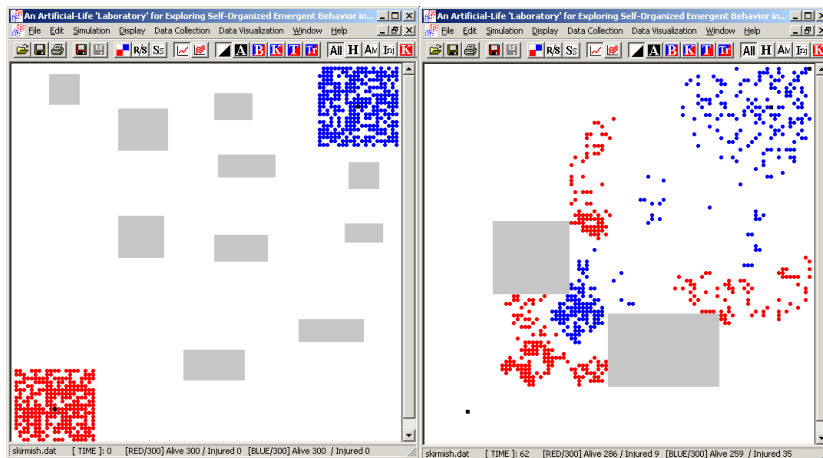


Notice that, compared to figures 14 and 15, the Red force is able to avoid heavy losses during the battle and the Red agents are able to get closer to the Blue flag. In figure 18, the complex Red movements of advance, retreat, and advance during the course of the scenario can be observed. The previous two examples again demonstrate that small changes in only a few parameters can lead to very different behaviors for the groups of agents, and highlight how complex behavior can arise in this agent based combat model by using just a few simple rules defining the personalities of the agents.

Adding terrain elements

The dynamics of the simulations can be changed further by adding terrain elements as described earlier. For example, the user could add impassible barriers either randomly or at key areas in a battlefield to see how this affects the overall combat. Figure 19 shows examples of this.

Figure 19. Screenshots of terrain elements^a



- a. Adding terrain to the battle space can alter the dynamics of the battlefield. On the left, the Blue force is defending their flag against the Red force in rocky or mountainous terrain, reducing overall mobility. On the right, a choke point in the battlefield reduces the Blue force's ability to move back into the battlefield once the Red maneuvers around them.

For example, in the left pane, the Blue defenders from example 3 are defending a flag in a rocky or mountainous territory which affect the two force's ability to maneuver. The right pane in figure 19 shows the Red force using a 'choke point' to reduce the pursuing Blue force's maneuverability and increase the success of the Red force in moving into position between the divided Blue force to take the Blue flag. Further examples of the effects of terrain on scenarios can be found in other CNA Ground Combat Studies [22, 24, 27, 28].

Examples of other CA-based models

ISAAC and EINSTEIN have inspired many other cellular automata-based combat models. Reference [2] has a detailed account of these models. We include the excerpts for two of the models here.

CROCADILE (Conceptual Research Oriented Combat Agent Distillation Implemented in the Littoral Environment), developed by the Australian Defence Force Academy, is an “open, extensible agent-based distillation engine.” In contrast to EINSTEIN, it “incorporates a more realistic—albeit more complicated—projectile physics representation of its weapons. Munitions may be fired with specified speed and heading and a 3D collision detection algorithm is employed to detect when agents are hit or where explosions occur” [2].

MANA (Map Aware Non-uniform Automata) is an agent-based model developed by New Zealand’s Defense Technology Agency. It is largely based on the design of ISAAC/EINSTEIN. It has built-in agents to model non-combatants. In addition, it “exhibits many interesting and unique characteristics, including the ability to provide agents with memory maps of the locations of previously sensed enemies” [2].

Conclusions

This primer provides an elementary introduction to ground combat models in general, and EINSTEIN in particular. Ground combat models have evolved over the past four thousand years from ancient board games to modern-day computerized simulations. However, the essence of a model has not changed. Models are abstractions of the real world that allow us to study and better understand the real world.

Current DoD combat models are predominantly deterministic in design and often incorrectly use the Lanchester-type equations to calculate force-on-force attritions. We show that the Lanchester equations can model fairly complex battle dynamics (e.g., ambush in the Logarithmic Law example, and use of reserves and operational loss in the Morse and Kimball case). However, they include too many simplifying assumptions to remain valid in many of today's combat problems. Multiagent-based models, based on the mathematics of cellular automata, offer a novel approach to combat problems that analysts should not ignore.

The CNA Ground Combat Studies have shown successful applications of EINSTEIN including the 3:1 attacker to defender ratio, patrolling in formation, reconnaissance, security operations, ambush, and call for fire. EINSTEIN is constantly being updated, and new features are added. Our review of CNA ground combat studies and the brief tutorial of the software toolkit show only a small fraction of the rich features that are already available in EINSTEIN.

One of our objectives is to encourage interested readers to try out EINSTEIN. To facilitate this, we have provided several sample scenarios to show the various features of EINSTEIN.

This page intentionally left blank.

Glossary

ABM	Agent-based model
CA	Cellular automata
CASTFOREM	Combined Arms and Support Task Force Evaluation Model
CGF	Computer Generated Forces
CIM	CNA information memorandum
COM	Center of mass
CROCADILE	Conceptual Research Oriented Combat Agent Distillation Implemented in the Littoral Environment
DMSO	Defense Modeling and Simulation Office
DoD	Department of Defense
EINSTEIN	Enhanced ISAAC Neural Simulation Toolkit
EMS	Essentials of Modeling and Simulation
GA	Genetic algorithm
GCS	Ground Combat Studies
GCAM	General Campaign Analysis Model
IED	Improvised Explosive Device
ISAAC	Irreducible Semi-Autonomous Adaptive Combat
ITEM	Integrated Theater Engagement Model
JCATS	Joint Conflict and Tactical Simulation
JDAM	Joint Direct Attack Munition
JFCOM	Joint Forces Command
JICM	Joint Integrated Contingency Model
JMEM	Joint Munitions Effectiveness Manual
JSIMS	Joint Simulation System
LLNL	Lawrence Livermore National Laboratory
LOS	Line of sight
MANA	Map Aware Non-uniform Automata
MCCDC	Marine Corps Combat Development Command
MCWL	Marine Corps Warfighting Lab
M&S	Modeling and simulation

M&S CO	Modeling and Simulation Coordination Office
NSS	Naval Simulation System
OneSAF	One Semi-Automated Forces
RPG	Rocket Propelled Grenade
RSAS	RAND Strategy Assessment System
TACWAR	Tactical Warfare
TOW	Tube-launched, Optically tracked, Wire-guided
USMC	U.S. Marine Corps
VV&A	Verification, validation and accreditation

References

- [1] Paul K. Davis and Donald Blumenthal. *The Base of Sand Problem: A White Paper on the State of Military Combat Modeling*, 1991 (A RAND note N-3148-OSD/DARPA)
- [2] Andrew Ilachinski. *Artificial War: Multiagent-based Simulation of Combat*. World Scientific Publishing Co. Pte. Ltd., River Edge, New Jersey, 2004
- [3] U.S. Department of Defense, *Modeling and Simulation Glossary*, Dec 1997 (DoD 5000.59-M)
- [4] Jame F. Dunnigan. *How to Make War: A Comprehensive Guide to Modern Warfare for the Post-Cold War Era*, 3rd Ed, William Morrow and Company LLC. New York, 1993
- [5] Sean D. Naylor. *War games rigged? General says Millennium Challenge 02 'was almost entirely scripted'*, Army Times, 16 Aug 2002
- [6] Peter P. Perla. *The Art of Wargaming*, Naval Institute Press, Annapolis, Maryland, 1990
- [7] Andrew Wilson. *The Bomb and the Computer*, Delacorte Press, New York, New York, 1968
- [8] "Go" Encyclopædia Britannica. 2008. Encyclopædia Britannica Online. 12 Sep 2008 <<http://www.britannica.com/EBchecked/topic/236403/go>>.
- [9] Frederick William Lanchester. *Aircraft in Warfare: The Dawn of the Fourth Arm*, Constable & Company Limited, London, 1916
- [10] National Research Council. *Modeling and Simulation, Vol. 9 of Technology for the Navy and Marine Corps, 2000-2030*, National Academy Press, 1997

- [11] Inara Kuck. *Warfare Simulation: Status and Issues for Space*, Kirtland AFB, NM, Phillips Laboratory, Sep 1996 (PL-TR-96-1038)
- [12] Inara Kuck. *Warfare simulation: status and issues for space - an addendum*, Apr 2003 (AFRL-DE-TR-2003-1037, Pt. 1–5)
- [13] EMS, DoD M&S CO, 13 Sep 2008, <<http://ems.dmsomil/welcome.html>>
- [14] Modeling & Simulation Coordination Office, 13 Sep 2008, <<http://www.msco.mil>>
- [15] U.S. Army Program Executive Office for Simulation, Training, & Instruction. *OneSAF*, 13 Sep 2008 <<http://www.peostri.army.mil/PRODUCTS/ONESAF/>>
- [16] 705th Exercise Control Squadron, Kirtland Air Force Base, JCATS system overview, 12 Sep 2008 <<http://www.dmoc.kirtland.af.mil/capabilities/Nodes/jcats.htm>>
- [17] Lawrence Livermore National Laboratory, Nonproliferation, Arms Control & International Security Directorate, Q Division/Proliferation Detection & Defense Systems, Conflict Simulation Laboratory. *JCATS Algorithm Manual (DRAFT VERSION)*, 1 Mar 2006
- [18] James G. Taylor. *Lanchester Models of Warfare*, Volume I and II, Operations Research Society of America, Arlington, VA, 1983
- [19] James G. Taylor. *Force-on-Force Attrition Modelling*, Operations Research Society of America, Arlington, VA, 1980
- [20] Philip M. Morse and George E. Kimball. *Methods of Operations Research*, 1951 (OEG Report 54)
- [21] Andrew Ilachinski. *Cellular Automata: A Discrete Universe*, World Scientific Press, Sep 2001

- [22] Jonathan Schroden and David Broyles. *EINSTEIN Simulations of Squad and Fire Team Operations*, Feb 2005 (CNA Annotated Briefing D0011705.A1)
- [23] Marine Corps Association. *Guidebook for Marines*, 18th Revised Edition, Marine Corps Association, 2001
- [24] David Taylor, Christopher Schmahl, and Ahmed Hashim. *Ground Combat Study: Summary of Analysis*, Oct 2000 (CNA Research Memorandum D0002706.A1)
- [25] Paul K. Davis. *Aggregation, Disaggregation, and the 3:1 Rules in Ground Combat*, RAND, 1995
- [26] Trevor N. Dupuy. *Understanding War: History and Theory of Combat*, Paragon House Publishers, New York, 1987
- [27] Jonathan Schroden and David Taylor. *An EINSTEIN Simulation of a Conventional Ground Combat Scenario*, Mar 2005 (CNA Annotated Briefing D0010873.A3)
- [28] David A. Broyles and Wendy R. Trickey. *Modeling Dispersed Units in Open and Urban Environments with EINSTEIN, Volume I: Main Text*, May 2006 (CNA Research Memorandum D0014187.A2)
- [29] Andrew Ilachinski. Personal communication with Kai Wang, 18 Sep 2008
- [30] Gregory Cox. *EINSTEIN visits the Persian Gulf (via Monterey CA): A naval requirements war "gamlet" held at the Naval Postgraduate School, 11-15 Feb 2002*, Mar 2002 (Annotated PowerPoint Brief)
- [31] Andrew Ilachinski. *EINSTEIN: A Multiagent-Based Model of Combat, Review Chapter for Artificial Life Models in Software (Springer-Verlag, Publisher)*, Apr 2008 (CNA Research Memorandum D0018082.A1)
- [32] Andrew Ilachinski. *Enhanced ISAAC Neural Simulation Toolkit (EINSTEIN): An Artificial-Life Laboratory for Exploring Self Orga-*

nized Emergence in Land Combat; Beta-Test Users Guide, Nov 1999
(CNA Informational Memorandum 9506100010)

- [33] Andrew Ilachinski. *EINSTEIN: An Artificial-Life "Laboratory" for Exploring Self-Organized, Emergent Behavior in Land Combat*, Sep 2000 (CNA Research Memorandum D0002239.A1)
- [34] ISAAC/EINSTEIN: An Artificial-Life Approach to Land Combat, CNA 2008. 12 Sep 2008 <<http://www.cna.org/isaac/>>
- [35] Department of Defense. *DoD Modeling and Simulation (M&S) Verification, Validation, and Accreditation (VV&A)*, 13 May 2003 (DoD Instruction 5000.61)
- [36] Department of the Navy. *Verification, Validation, and Accreditation of Models and Simulations*, 19 Apr 1999 (Secretary of the Navy Instruction 5200.40)
- [37] Department of the Army. *Verification, Validation, and Accreditation of Army Models and Simulations*, 30 Sep 1999 (Department of the Army Pamphlet 5-11)

List of figures

Figure 1. Various dimensions to modeling	5
Figure 2. Use of different models	8
Figure 3. Lanchester Square Law chart	15
Figure 4. Lanchester Linear Law chart	16
Figure 5. Lanchester Mixed Law chart	17
Figure 6. Lanchester Logarithmic Law chart.	19
Figure 7. Modeling steps and VV&A	34
Figure 8. Agent behavior parameters window	38
Figure 9. The default weapons arsenal in EINSTEIN	40
Figure 10. Example 1: screenshot	43
Figure 11. Example 1: Average attrition data	44
Figure 12. Example 2: screenshots	45
Figure 13. Example 3: screenshots	47
Figure 14. Example 3: attrition data	48
Figure 15. Example 3: CoM distance from Blue flag	48
Figure 16. Example 4: screenshots time series.	49
Figure 17. Example 4: attrition data	50
Figure 18. Example 4: center of mass data.	50
Figure 19. Screenshots of terrain elements	51

This page intentionally left blank.

List of tables

Table 1.	Example 1: agent parameters.	42
Table 2.	Example 3: agent parameters.	46

This page intentionally left blank.

Distribution list

Marine Corps Warfighting Lab

Marine Corps Combat Development Command

Defense Technical Information Center

This page intentionally left blank.

