GOOD DECISIONS UNDER FIRE:

Human-Level Strategic and Tactical Artificial Intelligence in Real-World Three-Dimensional Environments

by

David Ezra Sidran

A paper submitted as part of the

Doctoral Comprehensive Examination

University of Iowa

2007

Approved by ______________________________
Chairperson of Supervisory Committee

____

____

____

____

Date ______________________________

ABSTRACT

GOOD DECISIONS UNDER FIRE:

Human-Level Strategic and Tactical Artificial Intelligence in
Real-World Three-Dimensional Environments

by David Ezra Sidran

Chairperson of the Supervisory Committee: Professor Alberto M. Segre
Department of Computer Science

The overarching goal of this research is to create a system that makes human-level tactical and strategic decisions within a computer wargame environment, which is reactive, and can learn, from experience or textbooks: specifically, digitized images of battlefield maps from the West Point Atlas.

We present here a collection of algorithms, with implementation, that we have used to create a test-bed application for the analysis and study of such an AI system in hostile real-world three-dimensional environments. These include algorithms for determining tactical lines, determining flanks of lines, algorithms for analyzing and quantifying friendly and hostile forces exerted upon units, implementation of the five basic offensive military maneuvers as defined by the U.S. Army Field Manual (3-21) and utilities for creating new scenarios. Collectively these constitute a tool-set for the systematic exploration of this subject.

TABLE OF CONTENTS

LIST OF FIGURES

ACKNOWLEDGMENTS

The author wishes to thank his advisor, Dr. Alberto M. Segre, for his insightful comments, suggestions and wisdom and Kelly for never giving up.

GLOSSARY

Agent. In the context of this paper a Non Player Character (NPC) controlled by an Artificial Intelligence.

Combat Resolution Table (CRT). A table or series of tables that are used to determine the effect of one unit firing upon another. CRTs frequently employ cross indices of unit types, strength, distance and terrain.

Computer Generated Forces (CGF). Computer Generated Forces (CGFs) are computer systems that emulate battlefield entities and units. {Lockheed :vii}

DARPA. The Defense Advanced Research Projects Agency (DARPA) is the central research and development organization for the Department of Defense (DoD).

DoD. The United States Department of Defense.

Front. 1. The lateral space occupied by an element measured from the extremity of one flank to the extremity of the other flank. 2. The direction of the enemy. 3. The line of contact of two opposing forces. 4. When a combat situation does not exist or is not assumed, the direction toward which the command is faced." {Joint Publication 1-02:224}

Line of Operations. 1. A logical line that connects actions on nodes and/or decisive points related in time and purpose with an objective(s). 2. A physical line that defines the interior or exterior orientation of the force in relation to the enemy or that connects actions on nodes and/or decisive points related in time and space to an objective(s)." {Joint Publication 1-02:317}

Model. A physical, mathematical, or otherwise logical representation of a system, entity, phenomenon, or process. *References*: "Glossary of Modeling and Simulation Terms for Distributed Interactive Simulation (DIS)," August, 1995; DoD Directive 5000.59, "DoD Modeling and Simulation (M&S) Management," January 4, 1994 ; DoD 5000.59-P, "Modeling and Simulation Master Plan," October 1995 "M&S Educational Training Tool (MSETT), Navy Air Weapons Center Training Systems Division Glossary," April 28, 1994

For the purpose of this paper 'model' will be defined as a collection of functions (e.g. movement and combat resolution calculations) while a 'simulation' is a collection of data such as units, terrain maps, elevation maps and goal states. For example, a standard 'model' of Civil War battles could be used for different simulations of Gettysburg and Antietam.

OPFOR. Opposition forces; the enemy.

Order of Battle Table. (OOBT) A list of units; frequently organized in a hierarchal structure. An example: five subordinate companies in a battalion.

Range of Influence. The radius that a unit projects based upon weaponry and other factors. In TIGER this is adjustable for each unit type.

Schwerpunkt. (German) The "point of maximum effort" or the "center of gravity" of an attack. First used in describing the blitzkrieg maneuver.

Semi-Automated Forces (SAF). "Decisions in the wargame are made by human operators (Semi-Automated Forces)" {Lockheed :vii} Also see 'Tie Guy'.

Simulation. A method for implementing a model over time. *Reference*: DoD Directive 5000.59, "DoD Modeling and Simulation (M&S) Management," January 4, 1994. **Computer Simulation**: A dynamic representation of a model, often involving some combination of executing code, control/display interface hardware, and interfaces to real-world equipment. *Reference*: DoD 5000.59-P, "Modeling and Simulation Master Plan," October 1995. See also Model.

Tactical Problem Set. A 5-tuple that describes a state of a military simulation (see formal definitions Appendix 5).

TACWAR. A division level computer wargame used by the U. S. military for planning Desert Storm in 1991.

Tie Guy. A colloquial term for a defense contractor who makes tactical and strategic decisions and implements movements for units in a computer wargame. Tie Guys are used in "crewed simulations;" that is to say simulations in which most decisions are made by paid contractors; not AI. Also see 'Semi-Automated Forces'.

TIGER. Tactical Inference GenERator. The test-bed program created by the authors for this research.

Unit. A group of soldiers represented in the simulation by a single icon.

Victory Condition. In a scenario the method for determining the winner; ultimately, the definition of a scenario or Tactical Problem Set's 'goal state'.

World View. An individual unit's view of the map (filtered by a 3D line of sight algorithm). This may also be a headquarters' unit's world view which is the sum of its subordinate units' world views.

Chapter 1

INTRODUCTION

The first military simulation, or wargame, was probably *Wei Hai* (literally, 'encirclement') attributed to the legendary Sun Tzu at least 2,500 years ago.[1] {Perla 1990:16} Chaturanga[2], the Indian precursor to chess which employed a military theme, may be even older. {Falkener 1961:119-142} However, it was not until the 18th and 19th centuries that wargaming became a serious tool for military study. These wargames included John Clerk's use of model ships to investigate naval tactics in 1782[3] and G. H. R. J. von Reisswitz's *Instructions for the Representation of Tactical Maneuvers under the Guise of a Wargame* in 1824[4]. {Perla 1990:19-25} By the beginning of the 20th century the military service academies of the great powers employed wargames as both teaching devices and as simulations. (see *Figure 1* next page). {Perla 1990:61-103}

These wargames consisted of:

- Two sides (commonly labeled 'red' and 'blue') controlled by individuals or teams.
- A 'battlefield' or playing area that included detailed topography, elevation, roads, cities, etc.
- A set of rules that included tables for movement of units.

---

1 Some references put the age of *Wei Hai* at 5,000 years ago (Greenberg referenced in Perla p. 16) but this is probably far too early.

2 "Chaturanga" literally means "having four limbs (or parts)" and in epic poetry often means "army". The game reflects four-fold division of the ancient Indian army..." http://en.wikipedia.org/wiki/Chaturanga

3 John Clerk's "An Essay of Naval Tactics," 1782

4 *Anleitung zur Darstelling militarische manover mit dem Apparat des Kriegsspiels;* Kriegsspiels is traditionally translated as 'wargame' in English.

- A 'combat resolution table' (CRT) for calculating the results of red and blue units' fire.
- A set of scenario rules that include the Order of Battle Tables representing the units (red and blue) that are available in the scenario, a reinforcement schedule (a table that shows when and where units are available) and a set of Victory Conditions to determine the winner of the scenario.
- And, frequently, an umpire[5] that was the ultimate arbiter of disputes and interpreter of the rules.

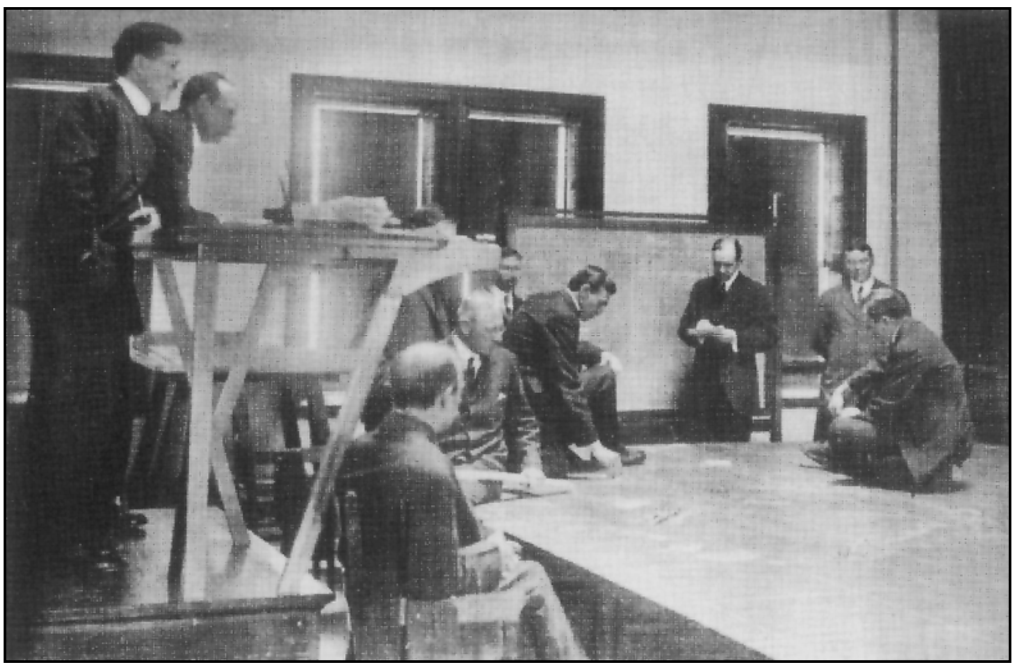


Figure 1 – A fleet tactical wargame being conducted in Luce Hall at the Naval War College circa 1914. (Photo courtesy of the U. S. Naval War College Museum).

[5] "Rear Admiral Ugaki, while presiding over the Japanese Staff pre-Midway wargame, overruled the dice throw of Lieutenant Commander Okumiya who was rolling to determine hypothetical bombing results on the Nagumo Carrier Force. Okumiya had determined that the Carrier Force had sustained nine hits and that both the Akagi and Kaga had been sunk. Rear Admiral Ugaki, however refused to believe that the pride of the Imperial navy could be so quickly swept away, and promptly resurrected both carriers; a *deus ex machina* that could not be repeated in the actual event some months later." {Sidran 1993:14} {Perla 1990:46}

The period after World War II was characterized by an explosive growth in the development of wargames for both the military and the hobbyist. Avalon Hill, the first company devoted to creating wargames for the general public, was established in 1954. Their first product was a board wargame entitled *Tactics*. {Perla 1990:115}

The first commercial computer wargame was *Tanktics* created by Chris Crawford in 1974. {Crawford 2003:187-203} This is a significant event in the development of computer wargames because *Tanktics* introduced, for the first time outside of the professional military community, the implementation of hidden movement, line of sight calculations and an artificial intelligence that controlled the opposing unit. *Tanktics* also employed a hexagonal map representation that was originally developed by the RAND Corporation (under contract to the U. S. military) and later taken up by Charles Roberts at Avalon Hill. {Perla 1990:114}

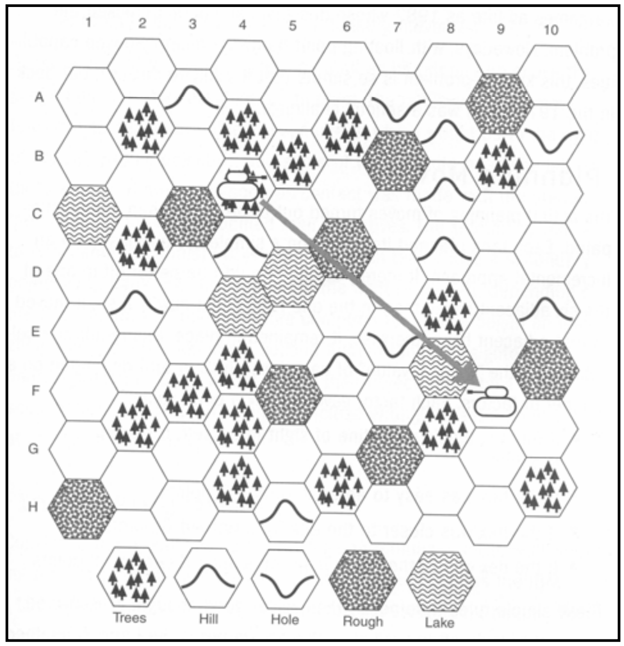


Figure 2 – A representation of line of sight calculations in *Tanktics*, 1974. Note: *Tanktics* did not have any graphical input or output. All I/O was via the terminal. {Crawford 2003:191}

The first use of a computer to support a wargame by the military probably occurred in 1956 when George Washington University supplied a mainframe to handle calculations for a Naval War College simulation. {Perla 2007:1}About the same time ATLAS[6], a NATO –

[6] "Atlas was one of many piston (because the algorithms were attrition based, for two forces just pushing at each other, as in a piston) games, a style that lasted into the 1980s." {Dunnigan email:1}

Warsaw Pact theater level simulation was developed by the U. S. army. {Dunnigan 1992:178} In the United Kingdom the *Action Information Organisation Tactical Trainer*, a "real time simulator" dated from the 1960's. A version running on a Ferranti Argus (the original Argus was an 8-bit machine developed in England for military use) was installed in Australia in 1975. {Ryan: 1} By 1977 staff officers in the U. S. V Corps were using *First Battle* as a corps training simulation exercise. {Robel 2007:1} Computer wargames that were used during the planning stages for Operation Desert Storm in 1991 included TACWAR[7], TAM[8] and JTLS[9]. {Dunnigan 1992:238-259}

Currently, the 'state of the art' in Computer Generated Forces (CGF) military simulations is the OneSAF project. SAF is an acronym for Semi-Automated Forces which means that defense contractors, or 'tie guys' (so called because of their ubiquitous neckwear), make all the decisions for the opposing (i.e., red) forces. {Lockheed 1998:vii}

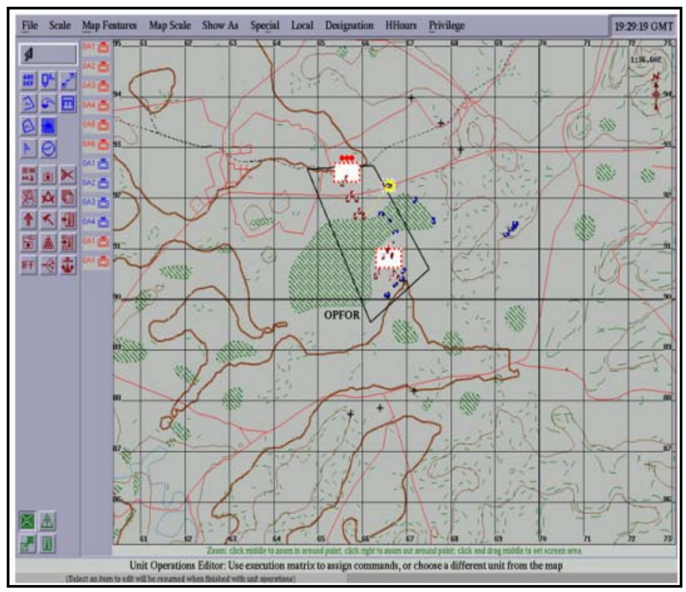


Figure 3 – Screen shot of OneSAF 'Unit Operation Editor'. Note red and blue units on toolbar in the left portion of the screen, terrain and elevation overlays and OPFOR units label. http://www.idealliance.org/papers/dx_xml03/papers/05-04-04/05-04-04-fig1.png

7 TACWAR was a divisional level wargame.

8 TAM was an operational level wargame designed by Mark Herman and Jim Dunnigan.

9 JTLS an updated version of the McClintock Theater Model wargame designed by Jim Dunnigan.

Chapter 2

TACTICAL & STRATEGIC DECISION MAKING

The overarching goal of this research is to create a system that makes human-level tactical and strategic decisions within a CGF environment, which is reactive, and can learn, from experience or textbooks (specifically digitized images of battlefield maps from the West Point Atlas).

Let us first consider two classic strategic maneuvers that are still studied at West Point: Grant's Vicksburg campaign and MacArthur's landing at Inchon. Grant's initial position was northwest of, and across the Mississippi River from, the city of Vicksburg His goal was not just the capture of the city of Vicksburg but the destruction or capture of all enemy forces that defended the city as well. To achieve this goal Grant moved his army south of Vicksburg and then, in a series of rapid maneuvers, northeast to Jackson and then west to Vicksburg in a sweeping envelopment maneuver.

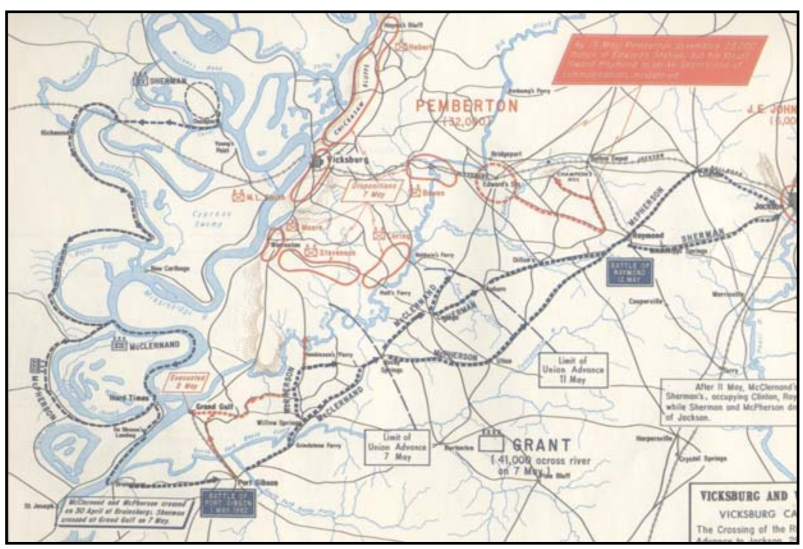


Figure 4 – Grant's Vicksburg campaign; May 1863. Source: {Griess, Atlas of the American Civil War, Map #20; 1986}

MacArthur in September 1950 was confronted with a dangerous strategic position: his forces were surrounded and confined to a small area around Pusan in the southeast corner of the peninsula. His solution involved an amphibious envelopment maneuver.

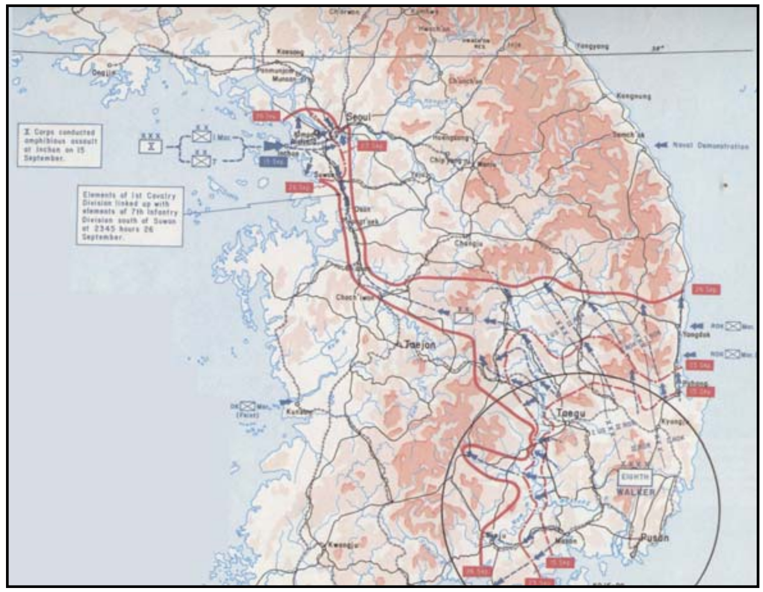


Figure 5 – MacArthur's landing at Inchon. Source: {Griess, Atlas for the Arab-Israeli Wars, the Chinese Civil War and the Korean War; Map #29 1986}

Why is creating a human-level artificial intelligence capable of making strategic decisions of this caliber such an extraordinary difficult problem? Steven Woodcock writes on the subject of strategic AI that what, "appears straight-forward enough to the player ('hey, they're over *there*, so I just need to move *this* and *this*')," is a daunting task because, "...computer AIs don't have the advantage of billions of years of biological evolution..." {Woodcock 2002:221}

Specifically, the problems of creating a human-level strategic AI include:

1. Amorphic and hard to formally define goal states. Consider the strategic situation in Figure 5: the UN forces were all but surrounded (only an open seaport allowed the troops to be supplied). MacArthur's strategic objective was to relieve pressure on the so-called 'Pusan Perimeter' by any means possible. His solution was an amphibious invasion far behind enemy lines that placed forces across the enemy's supply line. This strategy also included 'feints' or fake amphibious landings further south and an armored cavalry thrust from Pusan to the beachhead at Inchon. How would this goal state be formally described? The

strategic situation that Grant faced in 1863 (see Figure 4) was even more amorphous: he had to cross a river, capture a city, clear the Mississippi for navigation from Cairo to New Orleans and defeat two enemy armies. How would this goal state be formally defined?

2. How does an AI identify strategic or tactical opportunities? What constitutes a strategic or tactical opportunity? How do we analyze a potential opportunity? If we have a formal method of analyzing and ranking one situation (or state) over another, can use this to identify opportunities?
3. Just as chess AI is much more difficult than checkers AI, strategic and tactical AI is much more difficult than chess AI. The number of variables that describe the capabilities of the units (or pieces) is greater, the scale of the board (or map) is greater, and the number of potential states is substantially greater. Indeed, an exhaustive analysis of every potential state is not practical. Consequently, what appears to be a maxminimization problem {Osborne 2004:361-2} cannot be solved (with these methods) to optimality.

Strategic problems, by definition, involve larger maps, more units and unique problems such as supply lines and political objectives. Strategic problems:

1. Have more degrees of freedom than tactical problems.
2. Often have ill-defined goals or goals that are hard to formally define.
3. May have cascading and unforeseen effects such MacArthur's crossing of the 38th Parallel which directly led to China entering the Korean conflict and created "an entirely new war."[10]

[10] MacArthur cable to Washington on November 28. 1950, "We face an entirely new war."

4. May have multiple goals – consider Lee's invasion of the North in 1863 in which his strategic goals included encouraging "foreign intervention", "threatening Washington", "relieving Vicksburg" and supplying his troops {Longstreet:327-328} – which results in a Pareto optimality problem.

Are tactical problems easier to solve? The answer is almost certainly, 'yes'.

Consequently, we have divided our stated goal into three distinct research phases:

1. A Tactical Research Phase that consists of:
    a. Creating a multi-layered subsumption model (see Chapter 3) of active unit behavior in the small.
    b. Creating an environment-rich test-bed (TIGER) to implement the model of (a) (see Chapter 4).
    c. Modeling the five basic offensive maneuvers as specified in the U. S. Army Field Manual 3.21: Section II and that can be layered over the subsumption model of (a) (see Chapter 5).
2. A Learning Phase in which a Support Vector Machine (SVM) will be constructed (building on the existing TIGER program created in (1) above) that will take as input the set of West Point Atlas maps and expert analysis of the battles and will output a kernel that describes the strategic decision implemented by the analyzed tactical situation. {Hsu 2001} {Chang 2001}
3. A Strategic Research Phase that will apply the kernel of (2) using the tactical solver of (1) to new situations.

Our approach to part C of the Tactical Research Phase consisted of analyzing West Point Atlas maps and schematic diagrams of offensive maneuvers from FM 3-21.21, "The

Stryker Brigade Combat Team Infantry Battalion." We specifically chose this field manual because we were especially interested in conducting tactical research on the battalion level. We believe it is likely that results from the battalion level can be scaled up to larger formations.

For example, Rommel's tactics employed at Gazala (Figure 6) are clearly those of 'envelopment' as shown in FM 3-21.21 (see Figure 7). Compare the 'idealized' Envelopment Maneuver in Figure 7 to Rommel's actions in Figure 6 (note that in Figure 7 the offensive units are in blue while in Figure 6 they are red[11]). In Figure 6 the 'fixing forces' are the XXI and X corps (B4 and B3, respectively, on the map) while the 90th division, 15th division, 21st division and the Ariete division (A2 and B2 on the map) are the 'enveloping force'. Comparison of both Grant's and MacArthur's maneuvers to the canonical envelopment maneuver in Figure 7 shows that these are broad strategic envelopment maneuvers as well.

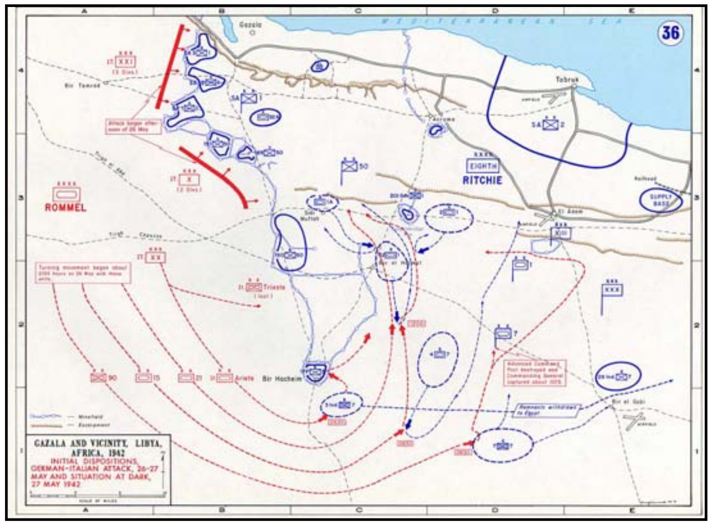


Figure 6 – Tactical situation Rommel versus Ritchie in North Africa in 1942. Source {Griess, Atlas for the Second World War, Europe and the Mediterranean; Map #36 1986}

[11] U. S. and allied units are traditionally displayed in blue; German units as red.

A precursor to recognizing these maneuvers is the ability to identify and plot offensive and defensive 'lines'[12] or 'fronts'[13]. We have implemented these methods and our techniques are described in Appendix 2.

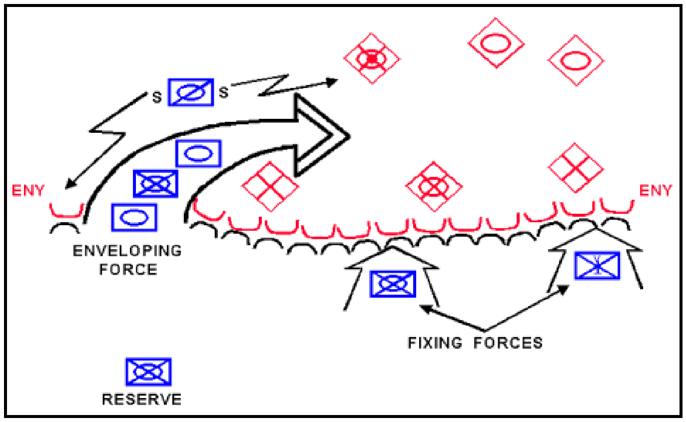


Figure 7 – The Envelopment Maneuver from U. S. Army Field Manual 3.21: Section II.

[12] "line of operations — 1. A logical line that connects actions on nodes and/or decisive points related in time and purpose with an objective(s). 2. A physical line that defines the interior or exterior orientation of the force in relation to the enemy or that connects actions on nodes and/or decisive points related in time and space to an objective(s)." {Joint Publication 1-02:317}

[13] "front — 1. The lateral space occupied by an element measured from the extremity of one flank to the extremity of the other flank. 2. The direction of the enemy. 3. The line of contact of two opposing forces. 4. When a combat situation does not exist or is not assumed, the direction toward which the command is faced." {Joint Publication 1-02:224}

Chapter 3

SUBSUMPTION, PLANNING, AND LEARNING SUCCESFUL TACTICS WITH A SUPPORT VECTOR MACHINE

Good tactical decisions must be built from the lowest levels up. The smallest units modeled in a CGF simulation are the foundation and building blocks of AI tactical decision making. Tactical decisions that are made at higher levels and are then imposed upon subordinate units without regard to the individual unit's situation, or state, will frequently fail as subordinate units are unable, or unwilling[14], to perform their assigned tasks.

Our design draws on Brooks' subsumption model in which information is 'passed up' from the lowest layers to the highest decision making layers. {Brooks 1985} In essence, subsumption architecture decomposes complex behavior into many simpler layers of increasingly more abstract behaviors. Each layer, in turn, can subsume the underlying layers' information. We implement this via a 'voting' method in which lower layers 'cast a vote' that is then factored into the next layer's decision.

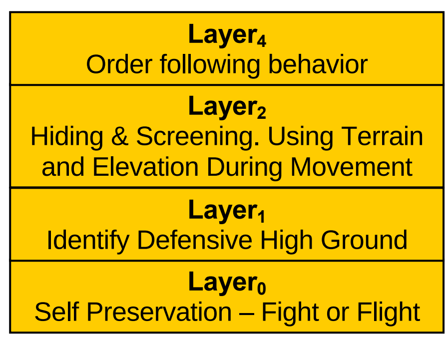


Figure 8 – Model of subsumption architecture for decision making.

The lowest layer, Layer$_0$ determines the "flight or fight" desire of a unit. We have implemented this layer (see Figures 44, 45, 46 and 47 for illustrations; see Appendix 1 for details. The output of this layer consists of a movement recommendation, represented as a vector (direction and magnitude). We

[14] Possible reasons that a unit may not be able to perform an assigned task include low morale, low fuel levels, difficult or impassable terrain between their current position and their objective, insufficient manpower, firepower or ammunition and blocking enemy units or waypoints controlled by an enemy units' range of influence.

have also outlined additional layers (see Figure 8) which will connect low-level unit behavior with the tactical decision making behavior described next.

It is imperative that any tactical planning mechanism be cognizant of the states, conditions and world views of the subordinate units that will be called upon to implement the plan (see Figure 9 for the relationship between lower level units and headquarters units with integration of the subsumption model).

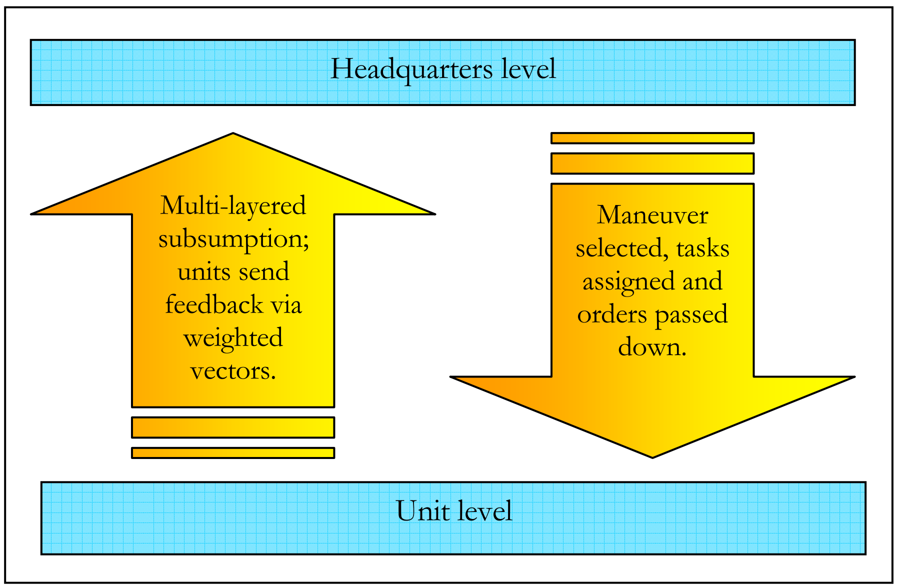


Figure 9 – Schematic diagram of the subsumption, decision-making and orders transmission cycle.

Previous investigations into planning techniques for CGF fall primarily into two categories: Case Based Reasoning (CBR) / Rule Based Deciding (RBD) {Ballard & Snively 2004} and various Bayesian Network schemes. {Yu}{Chia & Williams}{Penner & Steinmetz 2001}In both cases these techniques can be described as "top down." Furthermore, these methods do not implement learning techniques that draw upon the wealth of recorded, and analyzed, battles from history.

The lessons learned from historical battles are still applicable today. Consider Norman Schwarzkopf's "Hail Mary" maneuver (Figure 10) from the Desert Storm campaign (1991). It

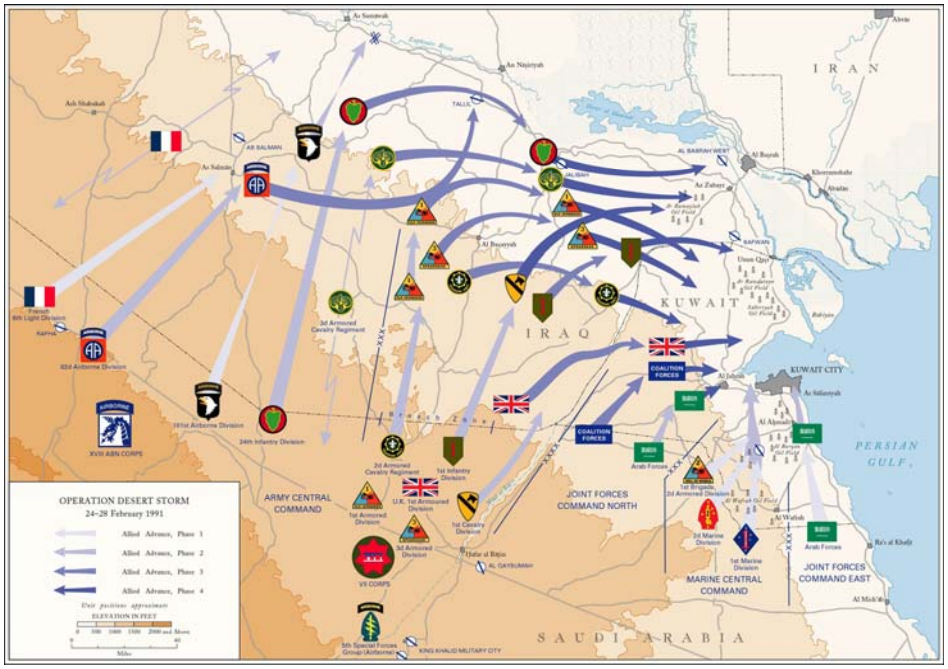


Figure 10 – General Norman Schwarzkopf's "Hail Mary" maneuver during Operation Desert Storm. Source: U. S. Military.

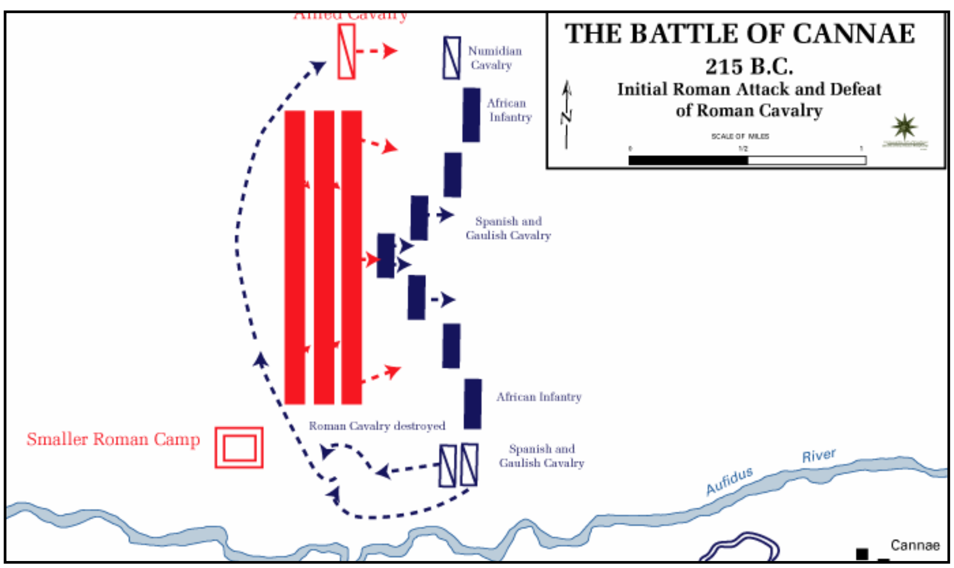


Figure 11 – The Battle of Cannae; Hannibal (Carthage) vs. Gaius Terentius Varro and Lucius Aemilius Paullus (Rome). Source: U. S. Military Academy, Dept. of History (West Point Atlas)

is the same envelopment maneuver previously seen implemented by Rommel in 1942 (Figure 6) and is instantly recognizable as the canonical envelopment maneuver from FM 3.21 (Figure 7). But the lineage of this maneuver is as old as warfare itself as shown in the battle of Cannae; 215 BCE (see Figure 11). We do not wish to discard this extraordinary database of military knowledge. Rather, we wish to implement a method that learns from it.

Chapter 4

TIGER: TACTICAL INFERENCE GENERATOR

TIGER (an acronym for Tactical Inference GenERator) is the test bed program that we have created for our research into tactical AI. It is written in C++ and runs on any Windows 98, XP or better computer.

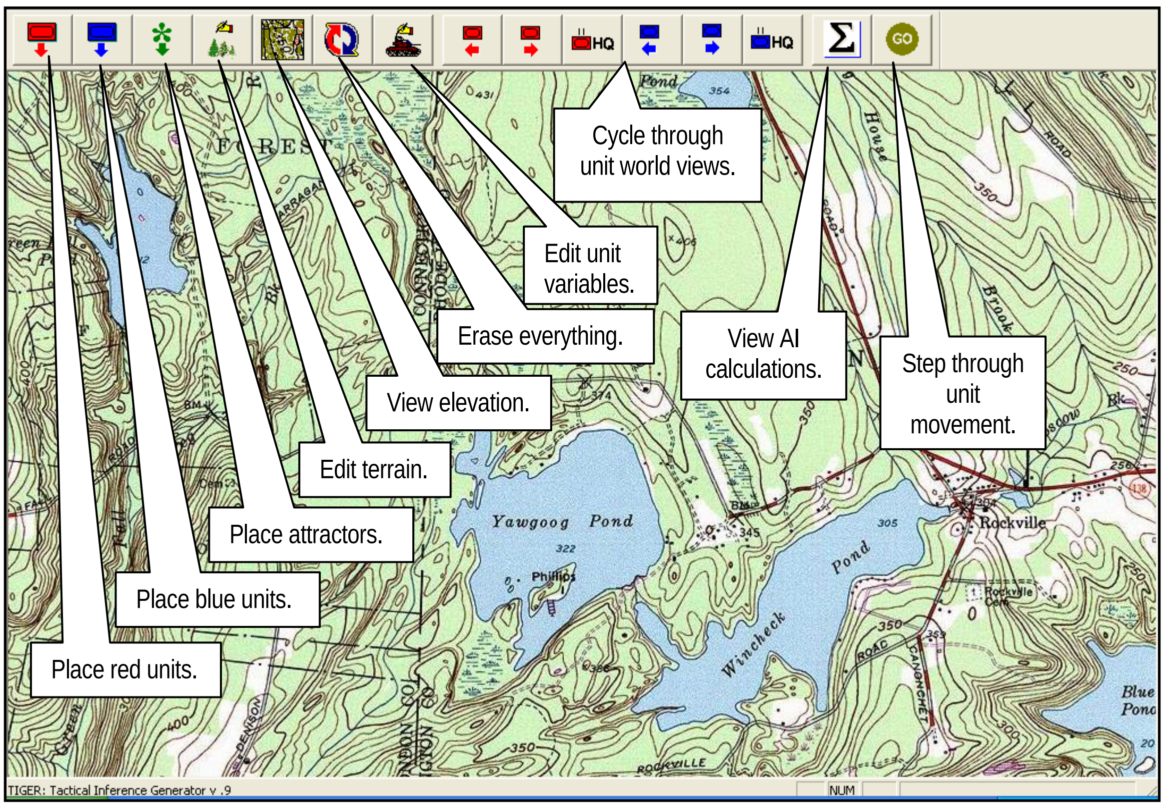


Figure 12 – TIGER main window functions.

The TIGER main window (see Figure 12 above) employs a floating, docking toolbar that allows access to the most commonly used features. These include placing red or blue military units, placing attractors, switching to terrain display overlay, switching to elevation display overlay, erase all units, edit the key unit variables, cycle through each unit's 'world view', or its superior headquarters' world view, view the results of tactical AI calculations and sequentially step all units through their movement cycles.

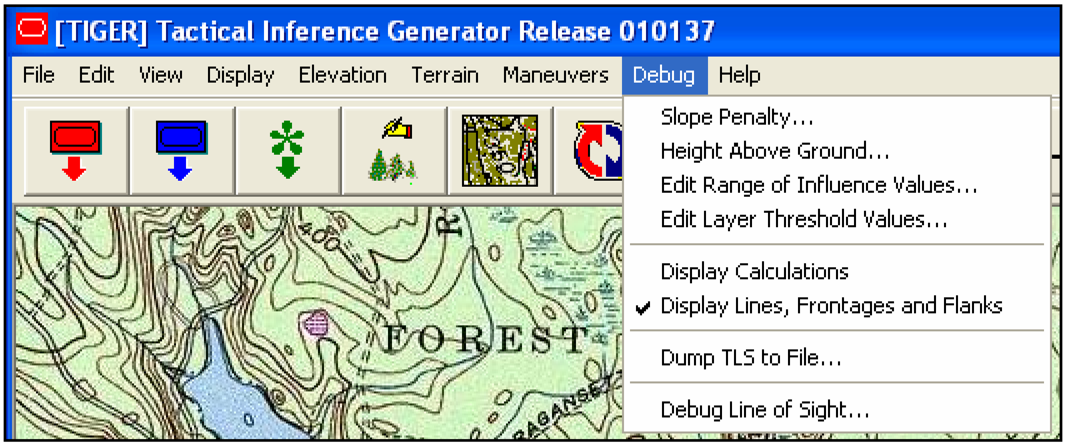


Figure 13 – TIGER pull-down menus.

Above TIGER's main window display are a series of pull-down menus that allow for loading and saving scenarios, selecting various viewing options (including viewing the entire map regardless of individual units' line of sight), loading and saving the terrain map, loading and saving the elevation map, forcing red or blue units to adopt specific tactical maneuvers, a debug menu that allows for setting certain variables or viewing visual output of certain AI functions and a help menu that displays information about the current TIGER build.

TIGER employs three overlays: a topographical map, an elevation map and a terrain map (see Figures 15 and 16, next page). To place a unit on the map, select the desired color (red or blue) and left-click the mouse at the desired location. The Unit Information dialog box will appear and the user can select the unit type (armor, mechanized infantry, cavalry, artillery or special). The user can enter the unit's strength, morale value and fuel in the dialog box. Right-clicking on a previously placed unit will bring up the same Unit

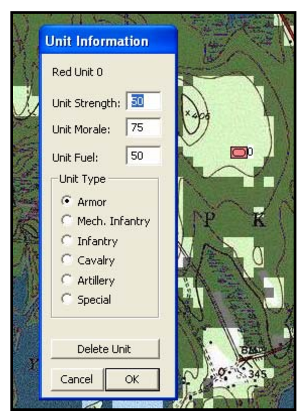


Figure 14 – Placing a unit in TIGER. Note: when a unit is placed the view is immediately changed to display the new unit's 'world view'.

Information dialog box which allows the user to change the values or delete the unit if desired.

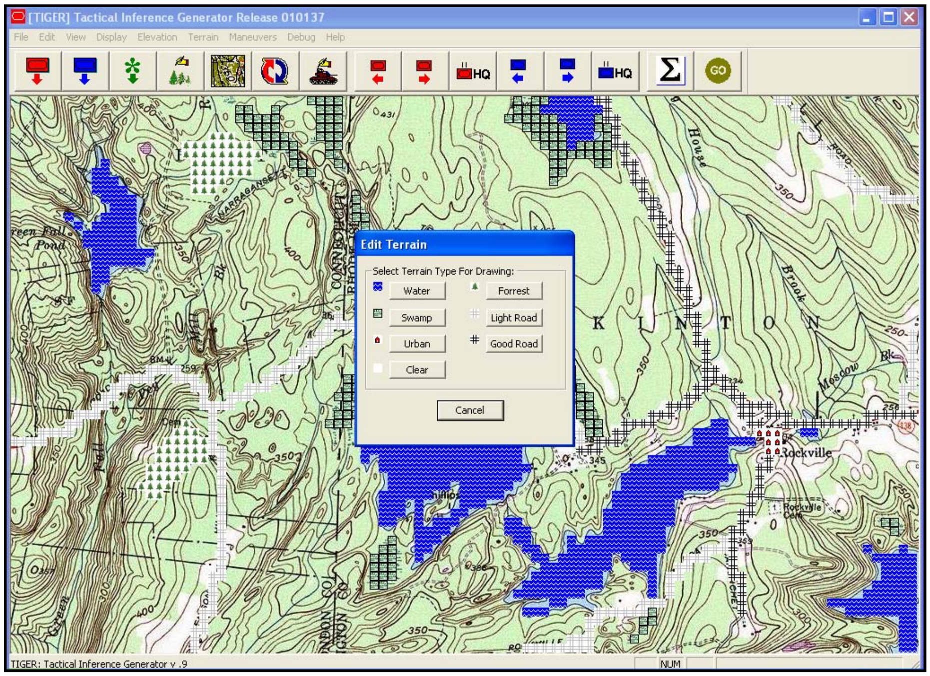


Figure 15 – TIGER terrain overlay view with terrain editing dialog box visible.

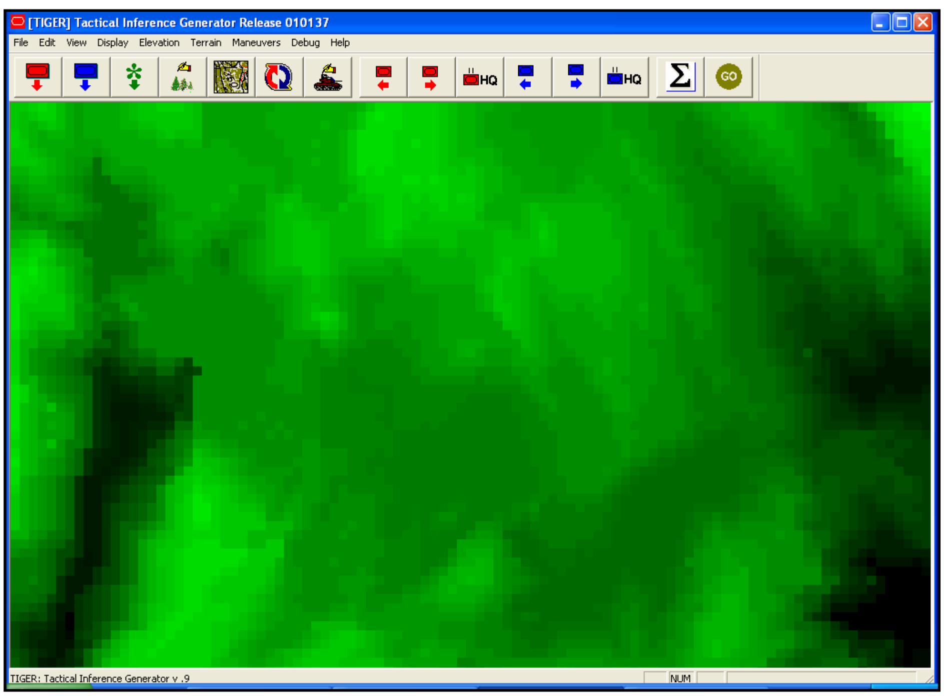


Figure 16 – TIGER elevation overlay view; darker colors represent lower elevation, lighter colors higher elevation.

The default mode for TIGER is 'line of sight' enabled (this can be turned off by clearing the elevation map). As units are added appropriate calculations are made using a three-dimensional Bresenham line algorithm. If the algorithm determines that an intervening elevation obscures the unit's line of sight the 'unobservable' areas are shaded a translucent gray and the areas (and the units located in them) are removed from that unit's calculations.

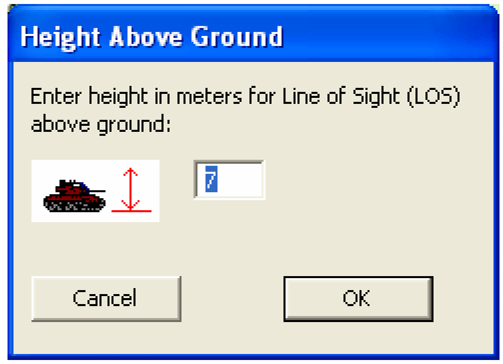


Figure 17 – Dialog box for adjusting the height above ground for Line of Sight calculations.

Figure 18 is a screen shot demonstrating line of sight. Blue Unit 0 can see Red Unit 0 but cannot see Red Unit 1. Consequently Blue Unit 0 is 'blind' to Red Unit 1 and its presence will not be used during BU_0 calculations. However, if another Blue unit is able to observe Red Unit 1 then this information is 'passed up' to the Blue Headquarters' World View and it will be used for BU_0 calculations.

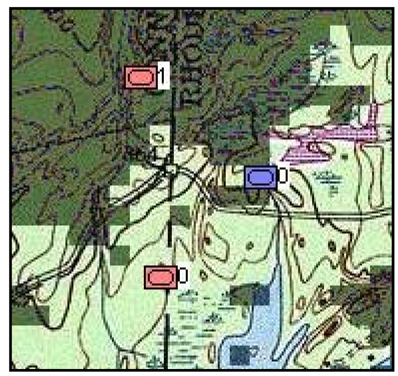


Figure 18 – Example of line of sight; note: Blue Unit 0 can see Red Unit 0 but cannot see Red Unit 1.

Attractors can be placed at any location by selecting the appropriate toolbar icon (see Figure 19) and then clicking on the desired location on the map. Attractors can have a value for Red or Blue units or both. Attractors that are labeled as an 'objective' are either used in $Layer_4$ calculations or may be placed by $Layer_4$.

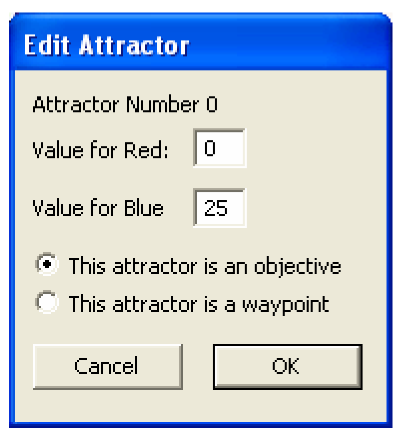


Figure 19 – The Edit Attractor dialog box. TIGER screen shot.

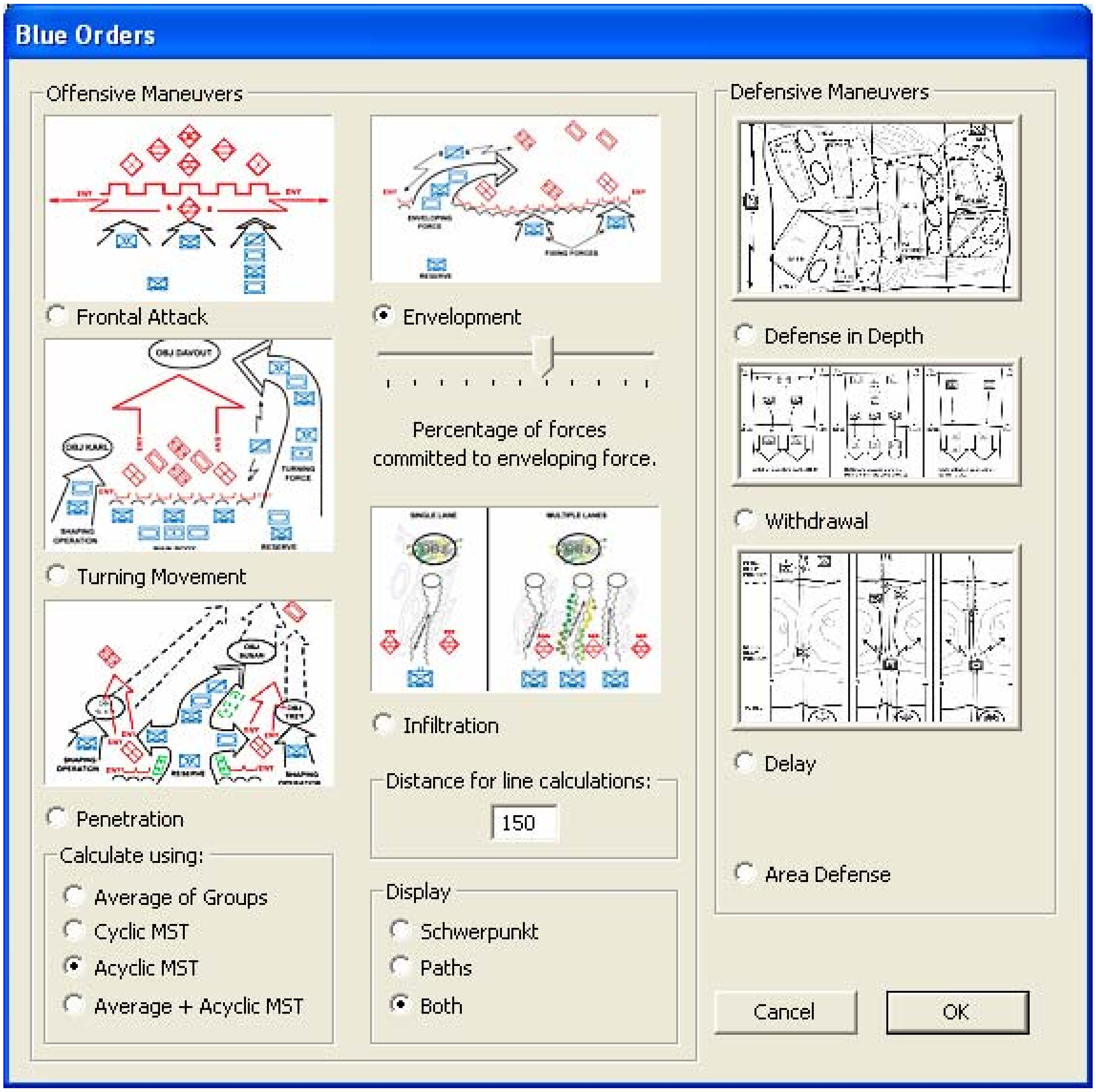


Figure 20 – Dialog box for selecting specific maneuvers. The user can specify s particular maneuver and calculation method. Other user defined variables include the distance for line calculations, percentage of forces assigned to the enveloping force and display options. TIGER screen shot.

Chapter 5

PLANNING

In 1975 Sacerdoti described how our perception of "plans as linear sequences of actions," is an illusion and that, "plans themselves are not constrained by limitations of linearity." He then introduces, "a new information structure called the procedural net that represents a plan as a partial ordering of actions with respect to time." "Basically, the procedural net is a network, of nodes, each of which contains procedural information, declarative information, and points to other nodes. Each node represents a particular action at some level of detail. The nodes are linked to form hierarchical descriptions of operations, and to form plans of action." Collectively, Sacerdoti, called this "monitoring system that uses a nonlinear representation of plans," NOAH; an acronym for Nets of Action Hierarchies. {Sacerdoti 1975: 206-7} NOAH falls into the category of Cooperative Distributed Planning (CDP) which, "proceeds through a hierarchy of plan levels, where at any plan level, a partial plan is a partial order of goals and primitive actions... (and) at each level expanding each unplanned goal by finding an applicable operator (called a SOUP procedure) that solves it." {desJardins et al. 1999:16}

Austin Tate, in an unpublished paper written in 1983, states that "all of the work on hierarchic non-linear planners has a root in Sacerdoti's landmark work on the NOAH planner." This includes Tate's own Nonlin which was introduced in 1977 and "extended the search space of Sacerdoti's NOAH to ensure that detected interactions were corrected for (NOAH only tried one of two legal plan re-orderings)." NONLIN employed "Goal nodes" that were "precisely specified by the planner and a Table of Multiple Effects (TOME) to “record all the world model changes at any node.” {Tate 1983} More recently Tate has developed O-Plan which “is a comprehensive planner. It is based on earlier work on Nonlin which is a hierarchical planning system able to generate plans as partially-ordered networks of activities and which can represent and check a variety of constraints for time, resources, etc.” {Tate et al. 2000}

Our planning problem, practically speaking, draws upon the previous work of Sacerdoti and Tate however we wish to extend it to include pattern recognition and learning. In Figure

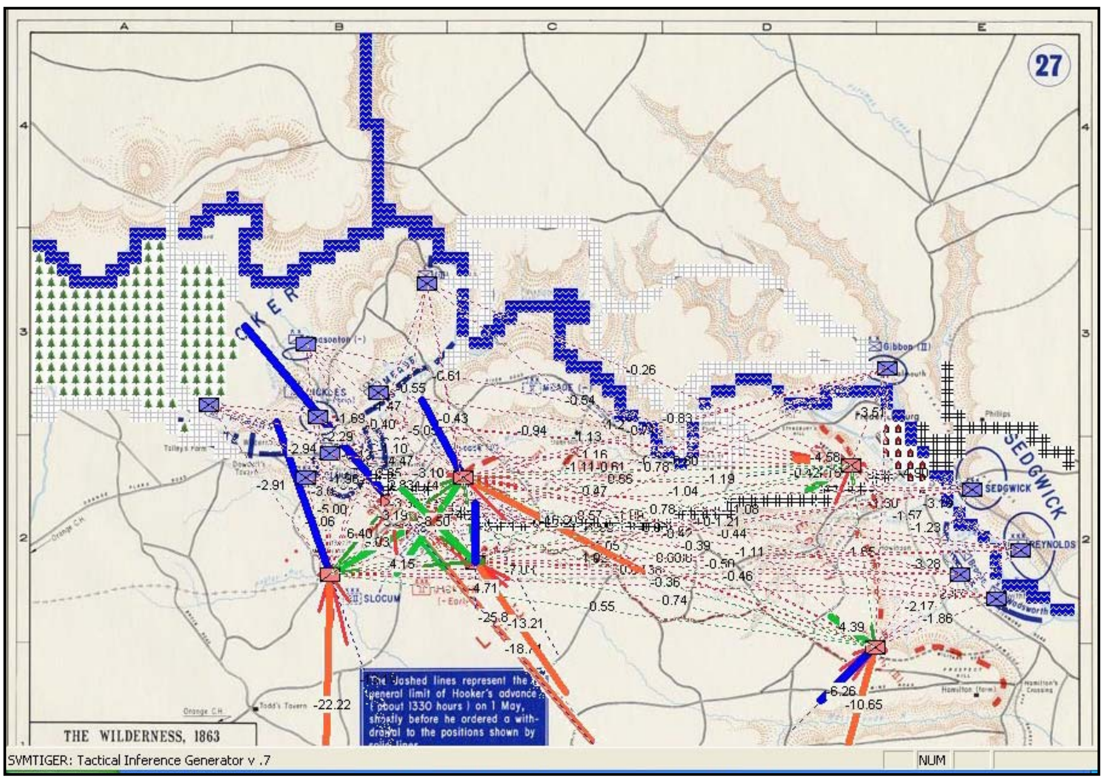


Figure 21 – The Battle of Chancellorsville, May 1863, imported into TIGER. TIGER screen shot.

21 we have imported a West Point Atlas map of the Battle of Chancellorsville into TIGER. This results in a unique pattern of units, lines of force (red, blue and green lines in Figure 21), terrain and elevation. Using the West Point Atlas as a database we will build up a library of these patterns that can be identified by SVM. Ultimately this database of patterns, and the historical results of the battles that these patterns represent, will be used to make tactical and strategic decisions.

TIGER also has the capability to store and plot the Range of Influence for each unit type (see Appendix 2) as well as to calculate lines and frontages and the ability to group units and to detect gaps in lines (see Appendix 2).

Figure 22 shows the logic that is currently employed by TIGER. This will be replaced by the SVM using features that are much richer.

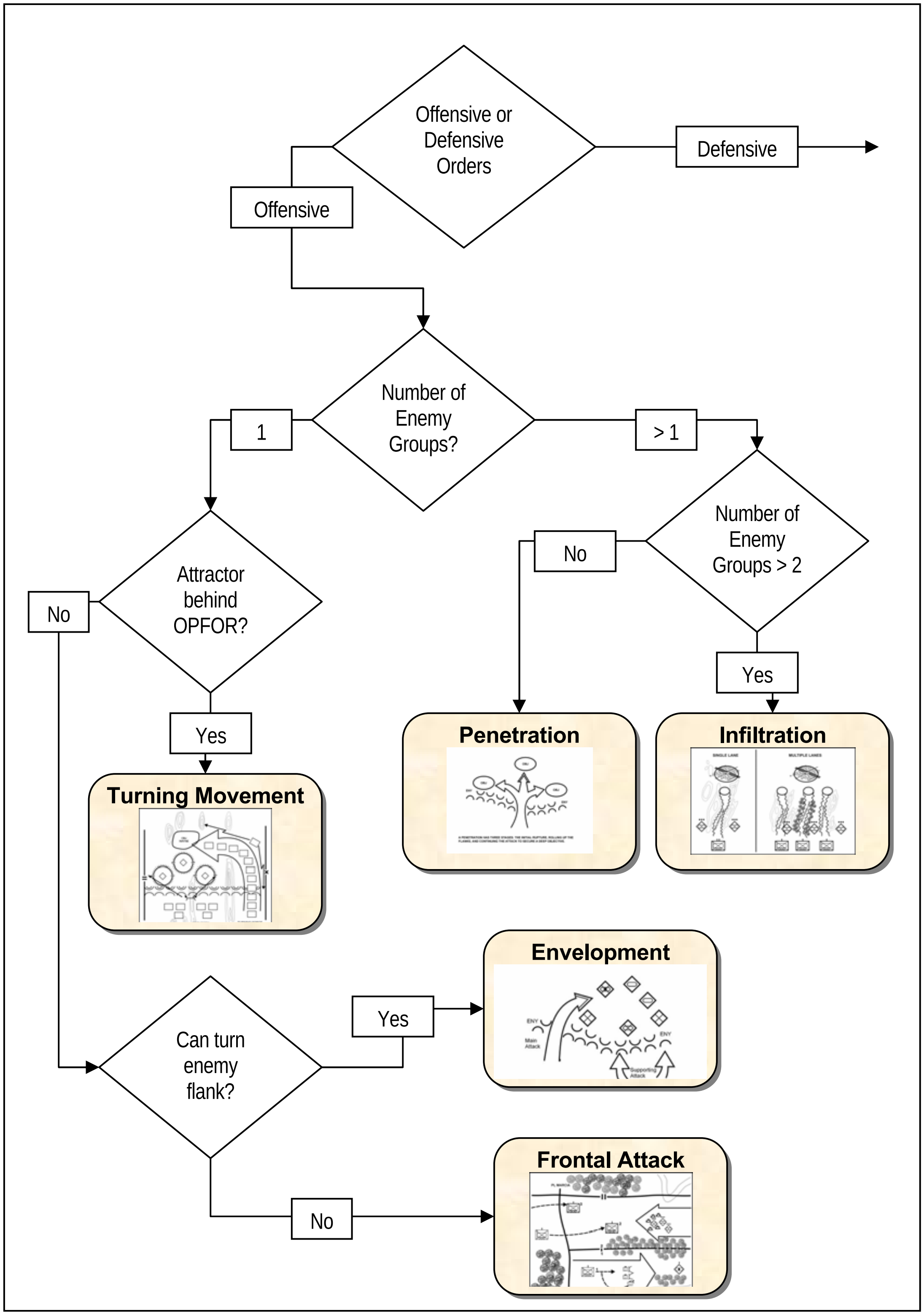
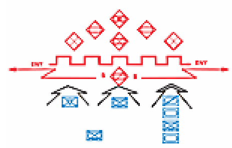
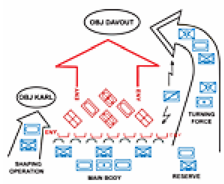
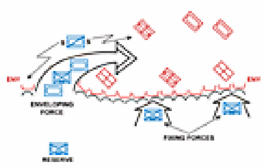
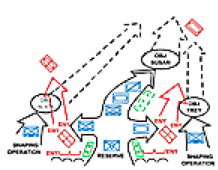
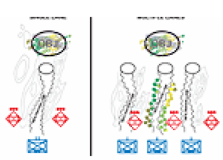


Figure 22 – Flow chart of offensive tactical decisions.

TABLE I: OFFENSIVE MANEUVERS AND OBJECTIVE ATTRACTOR PLACEMENT

Maneuver	How Attractor(s) placed
Frontal Attack 	TIGER places OBJECTIVE attractor at the average of the locations of the units in the OPFOR group.
Turning Maneuver 	USER places OBJECTIVE attractor.
Envelopment 	TIGER calculates OPFOR flanks (using MST) and places OBJECTIVE attractor for ENVELOPING FORCE. TIGER places OBJECTIVE attractor for FIXING FORCE.
Penetration 	TIGER places OBJECTIVE using MST calculations (see Appendix 2).
Infiltration 	USER places OBJECTIVE attractors.

TIGER IMPLEMENTATIONS OF OFFENSIVE FORMS OF MANEUVER

Below are examples of TIGER's implementation of the five canonical offensive maneuvers. Note: the algorithms currently used to implement these maneuvers may not be optimal. It is our intention to employ the SVM to identify and rank algorithms.

FRONTAL ATTACK

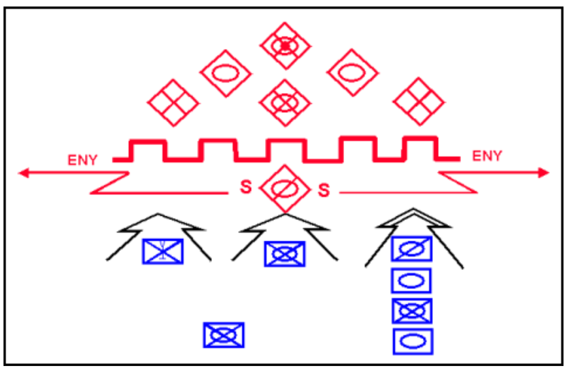


Figure 23 – Frontal Attack from Field Manual 3-21.

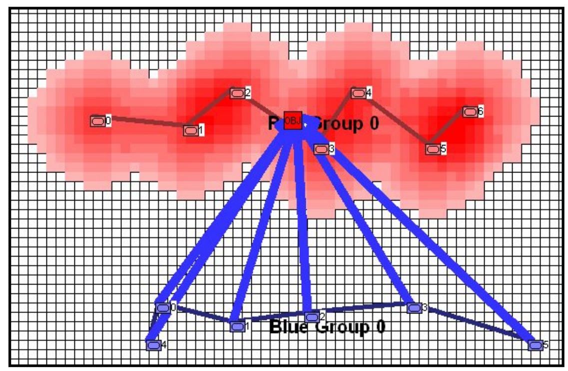


Figure 24 – Frontal Attack without terrain or elevation. TIGER screen shot. Schwerpunkt displayed.

The frontal attack algorithm is implemented by setting the OBJECTIVE attractor location to the average of the locations of units within the targeted Red Group. Waypoints are calculated using A* (with terrain and elevation if applicable).

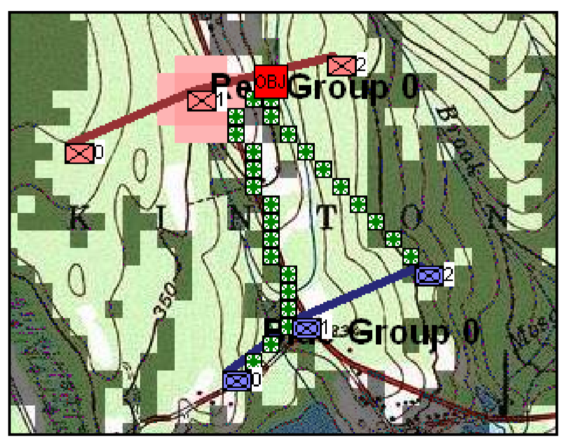


Figure 25 – Frontal Attack with terrain and elevation. TIGER screen shot. Waypoints displayed.

Future areas of research include modifying the algorithm to calculate the OBJECTIVE attractor as the point of lowest Red strength and/or lack of overlapping Range of Influence.

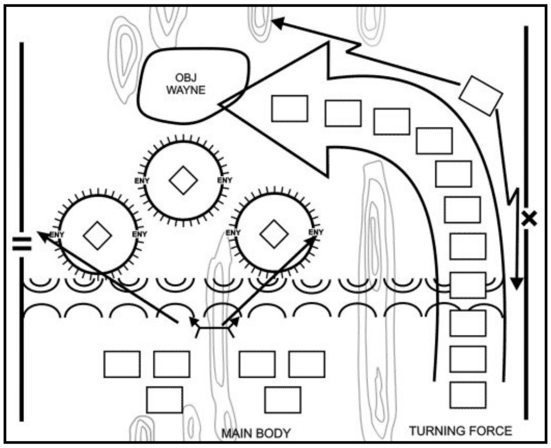


Figure 26 – Turning Movement from FM 3-21.

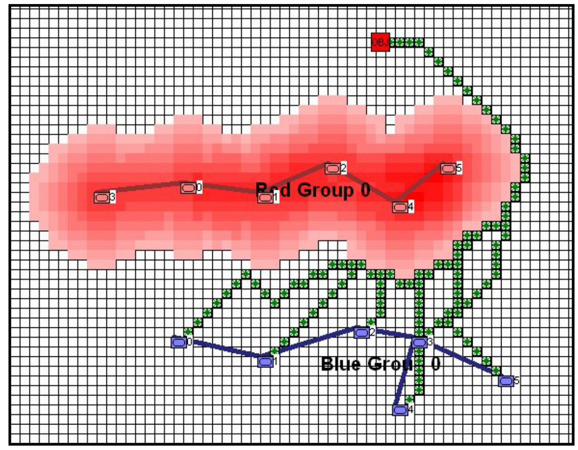


Figure 27 – Turning Movement without terrain or elevation. TIGER screen shot. Waypoints displayed.

The turning movement algorithm is implemented by the user setting the OBJECTIVE attractor. TIGER, using A*, calculates the lowest weighted path from each Blue unit to the OBJECTIVE. Note: in Figure 28 Red Unit 1 is not observable to any of the Blue units and, consequently, its Range of Influence is not displayed (nor considered by the A* pathfinding calculations).

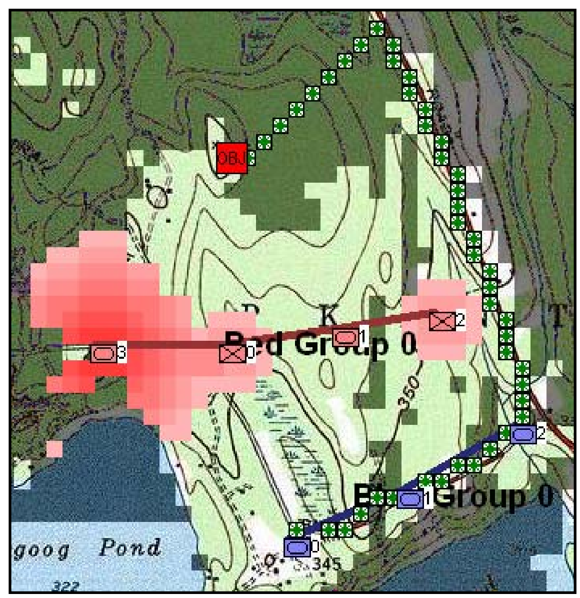


Figure 28 – Turning Movement with terrain and elevation. TIGER screen shot. Waypoints displayed. Note that Red Unit 1 is not observable to any Blue units.

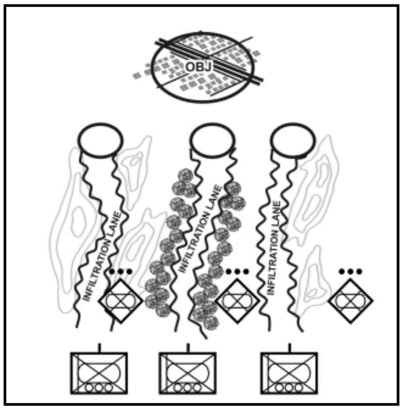


Figure 29 – Infiltration maneuver from FM 3-21.

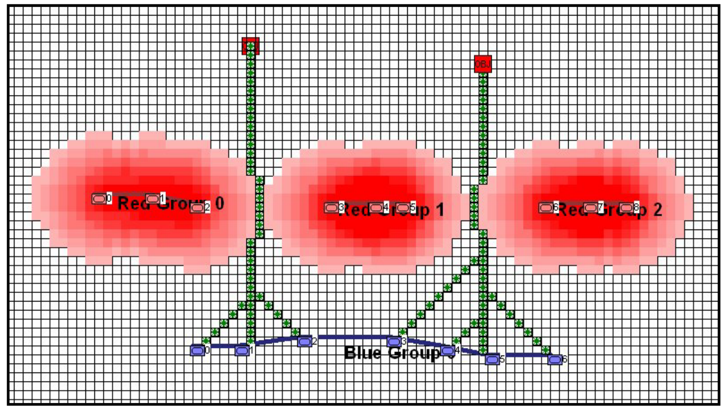


Figure 30 – Infiltration maneuver without terrain or elevation. TIGER screen shot. Waypoints displayed.

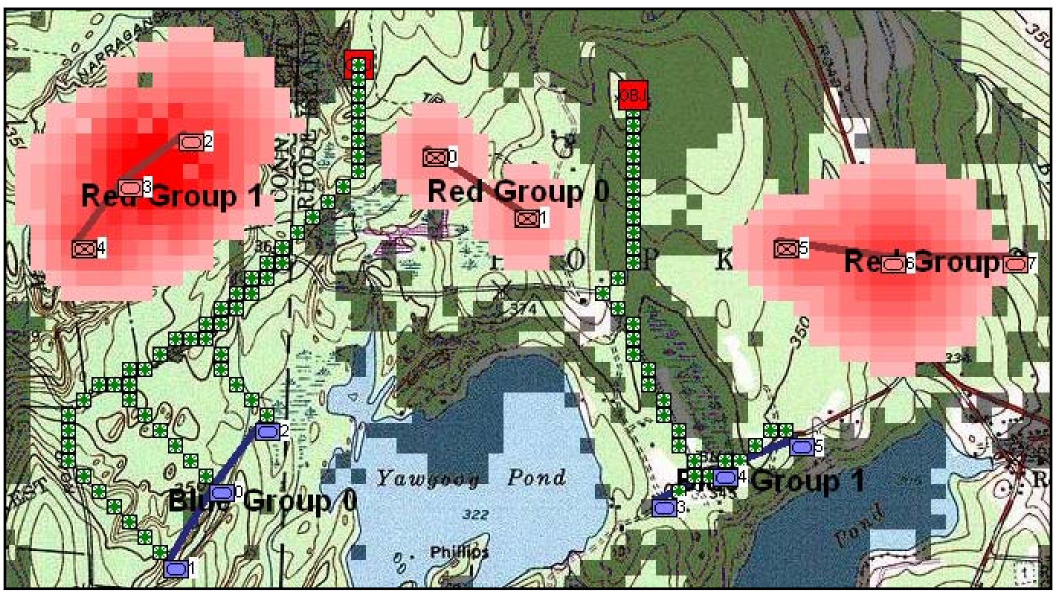


Figure 31 – Infiltration maneuver with terrain, elevation and range of influence. TIGER screen shot. Waypoints displayed.

The infiltration movement algorithm is implemented by the user setting two OBJECTIVE attractors. TIGER using A* calculates the lowest weighted path from each Blue unit to its nearest OBJECTIVE using terrain, elevation and range of influence in edge weight calculations.

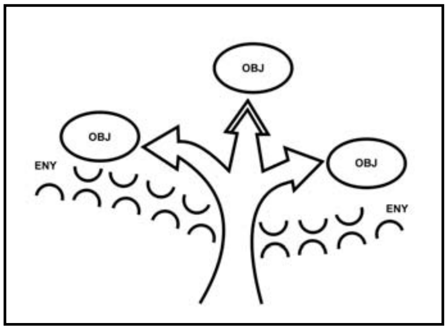


Figure 32 – Penetration maneuver from FM 3-21.

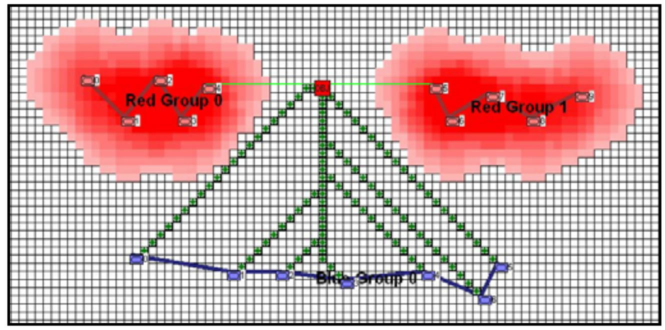


Figure 33 – Penetration maneuver without terrain or elevation. TIGER screen shot. Waypoints displayed.

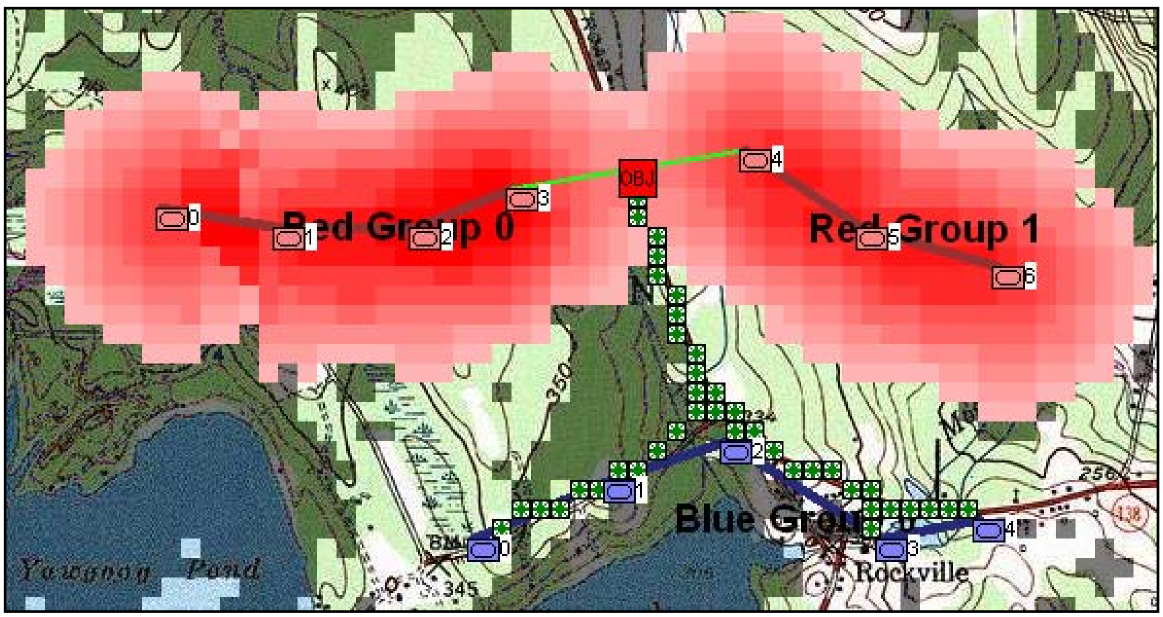


Figure 34 – Penetration maneuver with terrain, elevation and range of influence. TIGER screen shot. Waypoints displayed.

TIGER places the OBJECTIVE attractor by calculating the midpoint of the last weighted edge that was removed using Kruskal's MST algorithm (though other methods can also be selected Figure 20). TIGER using A* calculates the lowest weighted path from each Blue unit to the OBJECTIVE using terrain, elevation and range of influence in edge weight calculations.

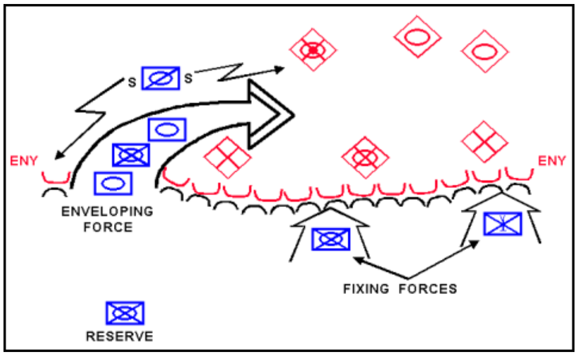


Figure 35 – Envelopment maneuver from FM 3-21.

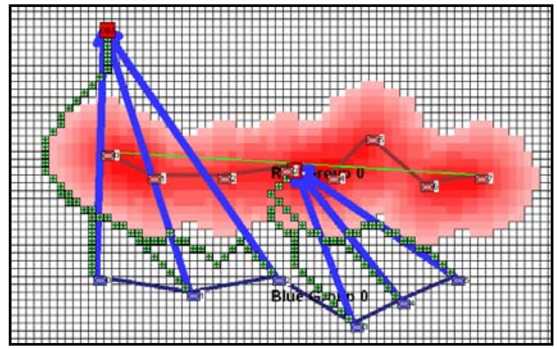


Figure 36 – Envelopment maneuver without terrain or elevation. TIGER screen shot. Schwerpunkts and waypoints displayed.

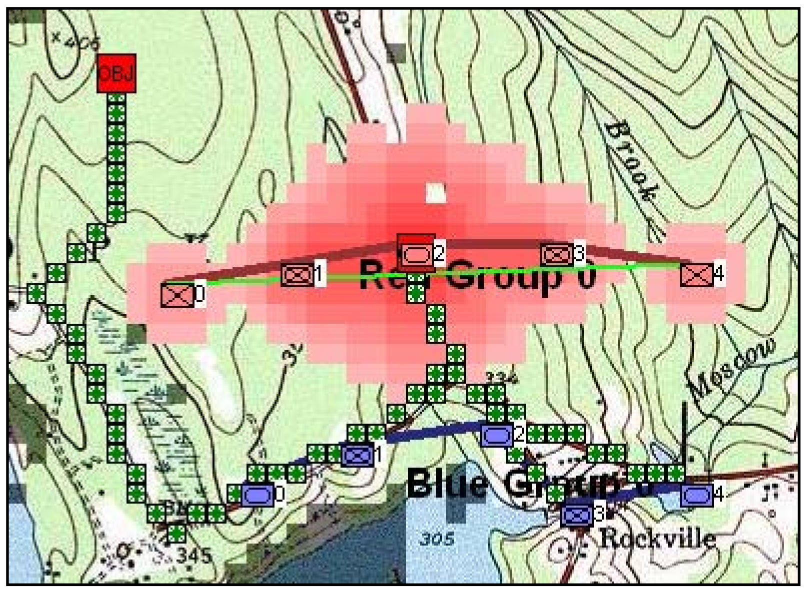


Figure 37 – Envelopment maneuver with terrain, elevation and range of influence. TIGER screen shot. Waypoints displayed. Note: the OBJECTIVE for the FIXING FORCE is under Red Unit 2.

TIGER places the ENVELOPING FORCE OBJECTIVE attractor by first calculating the flanks of the Red Group and then determining the closest flank to the Blue Group. TIGER then offsets the OBJECTIVE attractor by the distance from the center of the Red Group to the center of the Blue Group. TIGER places the FIXING FORCE OBJECTIVE attractor by calculating the center of the Red Group. The percentage of Blue units that are assigned to either the FIXING FORCE or the ENVELOPING FORCE is determined by the USER (see Figure 20). The default value is 60% of Blue units are assigned to the ENVELOPING FORCE. TIGER then assigns Blue units to either the FIXING FORCE or the

ENVELOPING FORCE based on location. TIGER using A* calculates the lowest weighted path from each Blue unit to the OBJECTIVE using terrain, elevation and range of influence in edge weight calculations.

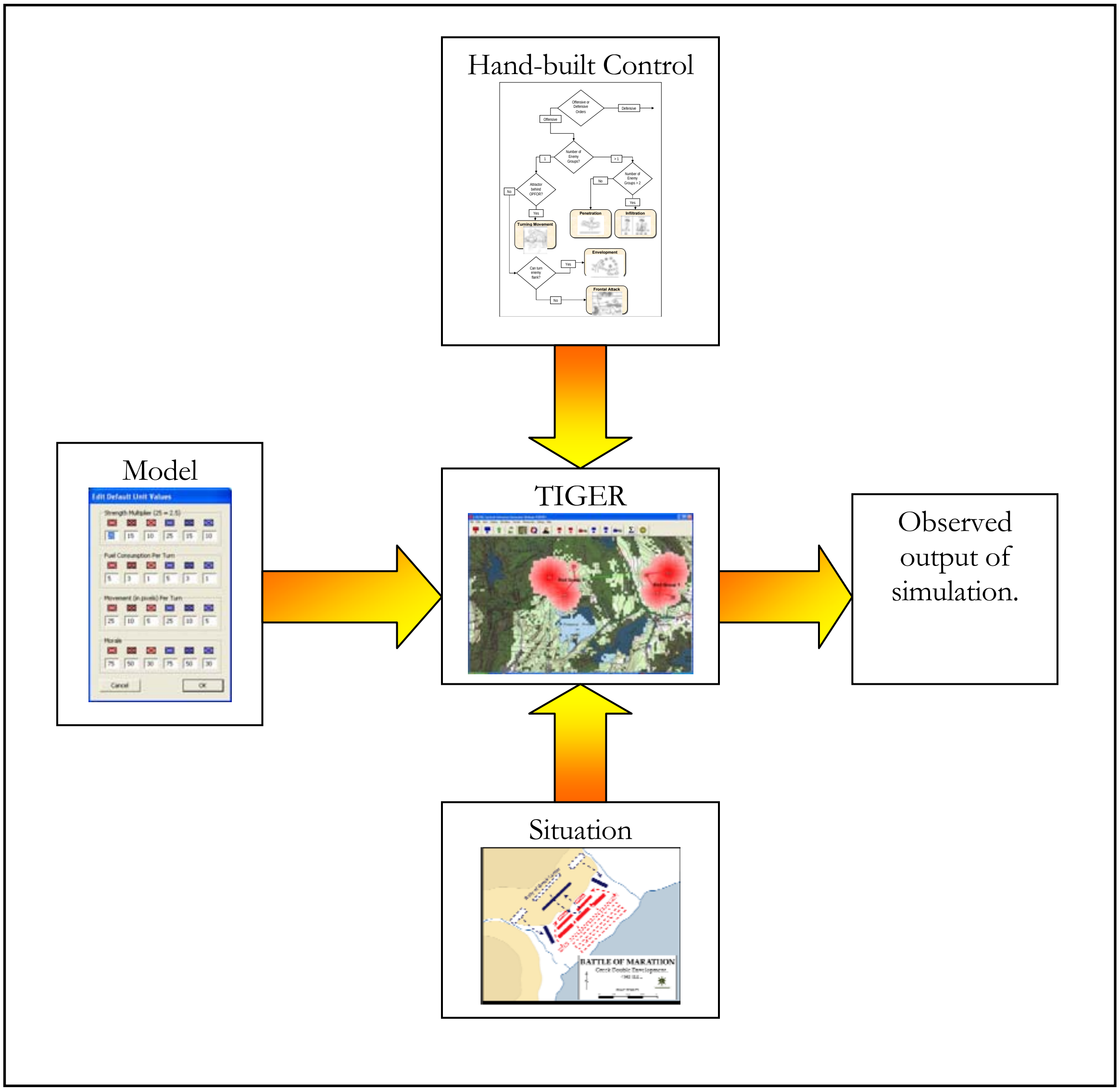


Figure 38 – "Hand-built" mode.

Chapter 6

FUTURE RESEARCH

It is our intention to continue our research with tactical maneuvers to include the four defensive maneuvers (Defense in Depth, Withdrawal, Delay and Area Defense). {Department of the Army 2003: Section 2}

We next intend to modify the existing TIGER test-bed program to import the entire collection of the West Point Atlas series of historical battles (see Figure 21 as an example). We will then manually add terrain, elevation and unit data constructing Tactical Problem Sets from each map. Support Vector Machines (SVM), though first reported by Vapnik in 1979 have only since the late 1990s become an important tool in pattern recognition and supervised learning research. SVMs have been used for handwriting recognition, object recognition, speaker identification, charmed quark detection and face detection in images. {Burges 1998:121} SVMs appear to be an excellent candidate for detecting the patterns of maneuver types and, when combined with data of the results of these maneuvers, will be able to learn and implement strategic planning over the tactical layers provided.

The input to the proposed SVM would consist of digitized West Point Atlas maps that have been prepared to include unit positions, strengths, morale levels, fuel levels, etc. as well as the matching terrain and elevation overlay files. The output from the proposed SVM would be a series of unit movement vectors accompanied by explanatory text.

We will then construct a SVM that will analyze the Tactical Problem Sets and the historical results with the intention of creating an SVM capable of learning tactics and strategy from historical Tactical Problem Sets.

We anticipate that training the SVM to recognize patterns produced by the historical Tactical Problem Sets will be the most difficult area of future research. While SVMs have been successfully employed for various kinds of visual recognition, including facial identification, this is a new application for an SVM.

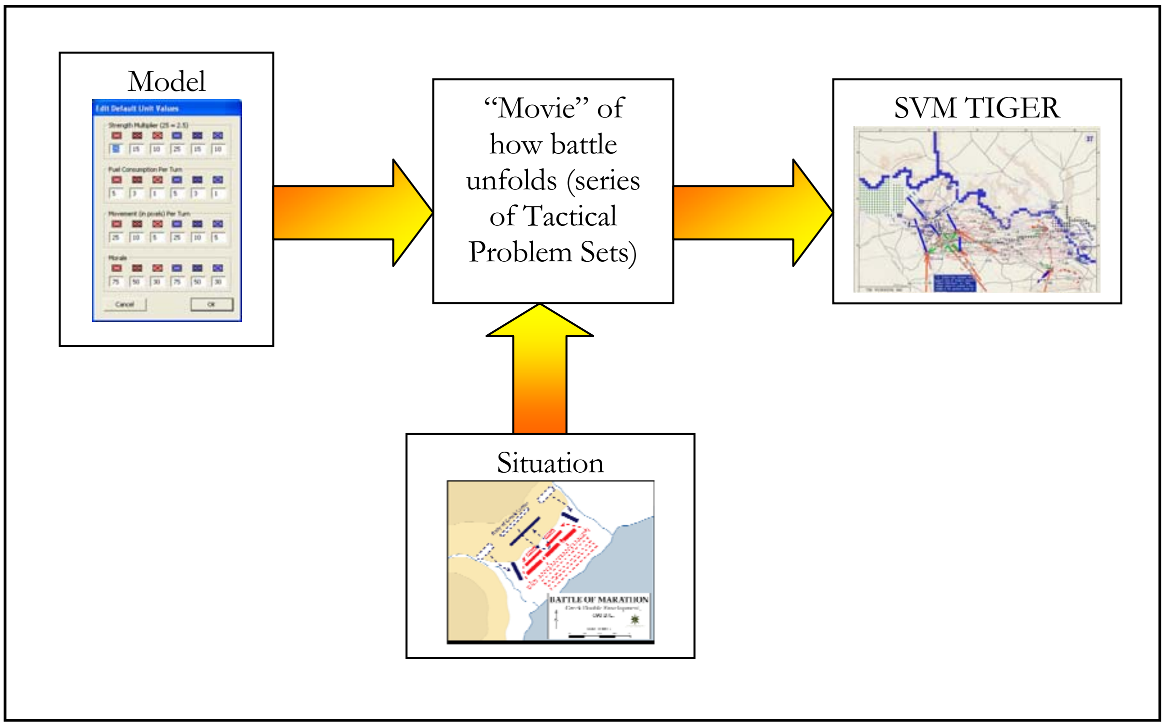


Figure 39 – "Learning" mode.

After the SVM is trained to recognize specific tactical and strategic positions using a method similar to the flow chart in Figure 22 it will produce a plan to exploit the tactical or strategic position to its advantage.

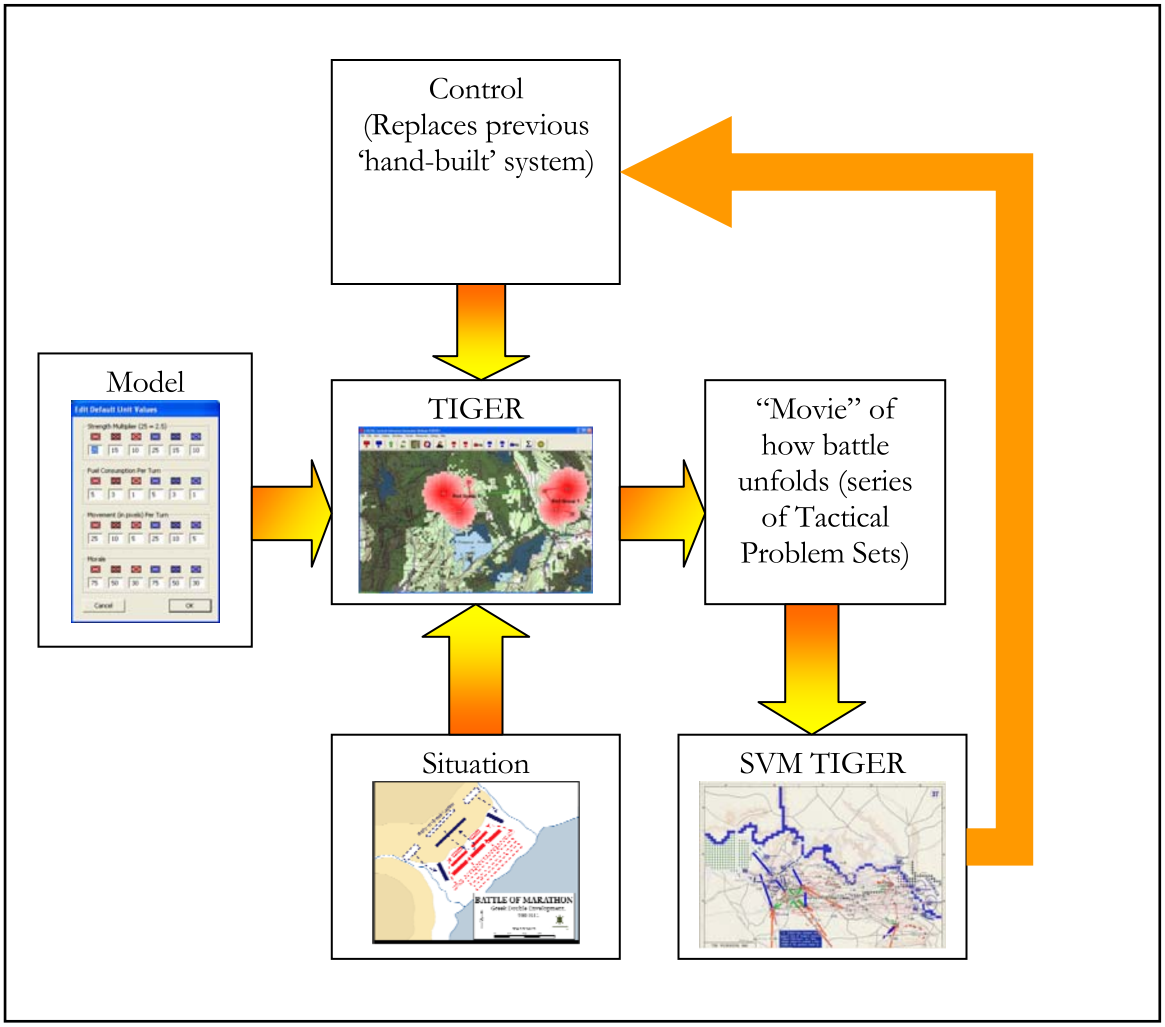


Figure 40 – Final System.

Appendix 1

$LAYER_0$: CALCULATING FORCES UPON UNITS

Here we present algorithms for determining the 'attitude' of a unit; i.e. if the unit wishes to assume the offensive, is capable of assuming the offensive, if the unit should move to the defensive or if it is fleeing in panic.

Algorithm For Calculating $Layer_0$ Weighted Vector:

1. For every unit:

2. Determine visible friendly units (using 3D LOS).

3. Calculate their 'magnitude'.

4. Calculate resultant vector.

5. Determine visible enemy units (using 3D LOS).

6. Calculate their 'magnitude'.

7. Calculate resultant vector.

8. Calculate sum of vectors and calculate magnitude'

Definition 1.1: *Interaction Magnitude* along vector A→B is calculated

((f*s*t)/d) *where*

1. Unit A = 'Active Unit'

2. Unit B = Unit under consideration;

if friendly, f=1, else f=-1

3. d = Euclidean distance between units in pixels / 10

4. s = Strength of Unit B.

5. t = Unit Type of B.

■

In figures 41 and 42 we demonstrate an example of the calculation of the attraction of two 'friendly' units. Note: the 'attraction' is mutual *but* Red 1 (armor) has a stronger *magnitude* than Red 2 (mechanized infantry).

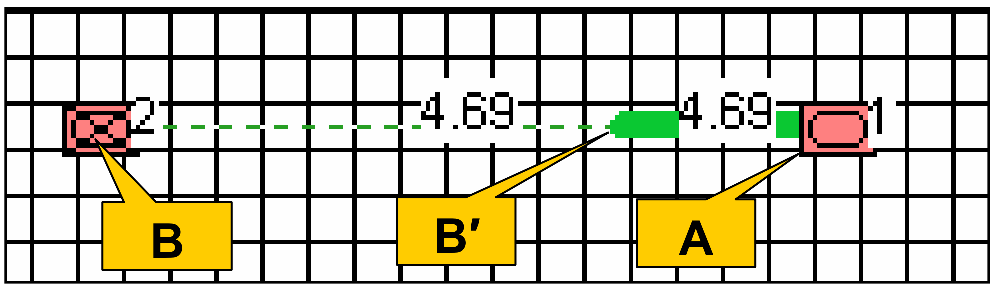


Figure 41 – Red Unit 1 is attracted to Red Unit 2 with a magnitude of 4.69. TIGER screen shot.

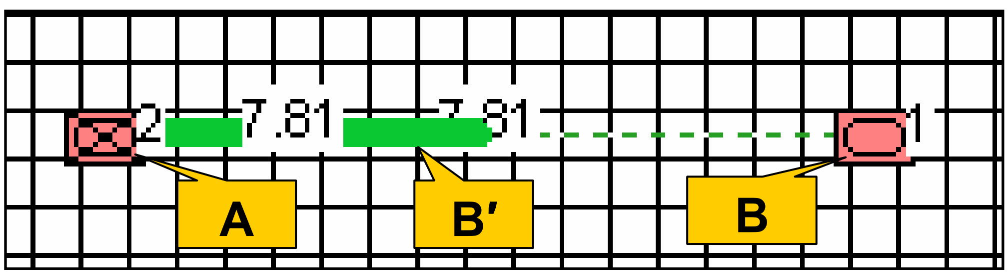


Figure 42 – Red Unit 2 is attracted to Red Unit 1 with a magnitude of 7.81. TIGER screen shot.

The reason that the Red armor unit has a greater magnitude than the Red mechanized infantry unit is because even though they are the same distance from each other and they have the same strength (default strength = 50), the strength multiplier is greater for armor than it is for mechanized infantry. The user can change these values in the Edit Default Unit Values dialog box (see Figure 43).

Friendly and hostile force vectors can be summed to create an overall force vector for each unit. This resultant force vector is the Layer$_0$ vector which can then be 'passed up' to Layer$_1$ as a 'vote' to determine the overall decision to assume offensive or defensive maneuvers.

In Figures 44, 45, 46 and 47 (below) we demonstrate the results of force vector calculations for various Red and Blue unit situations.

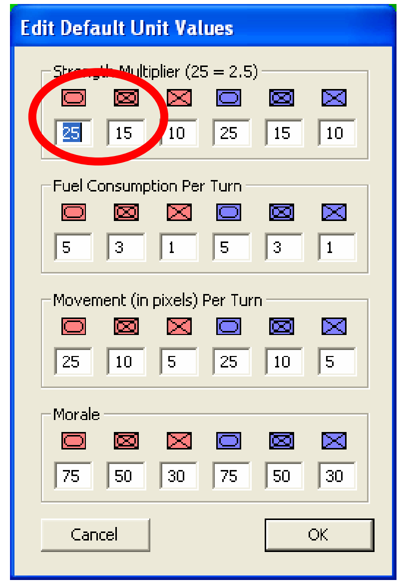


Figure 43 – The Edit Default Unit Values dialog box. In this example Red Armor units have a Strength Multiplier of 2.5 while Red Mechanized Infantry units have a Strength Multiplier of 1.5). TIGER screen shot.

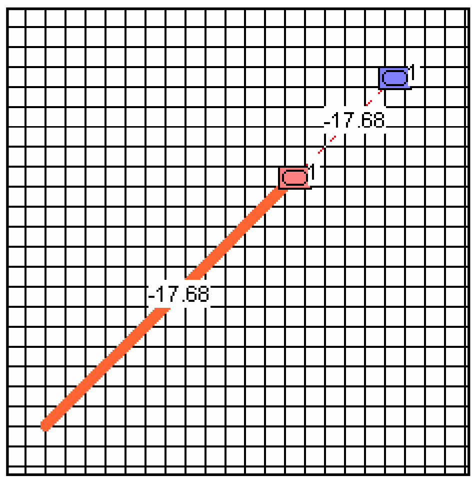


Figure 44 – Red Unit 1 'flees' from Blue Unit 1 – TIGER screen shot.

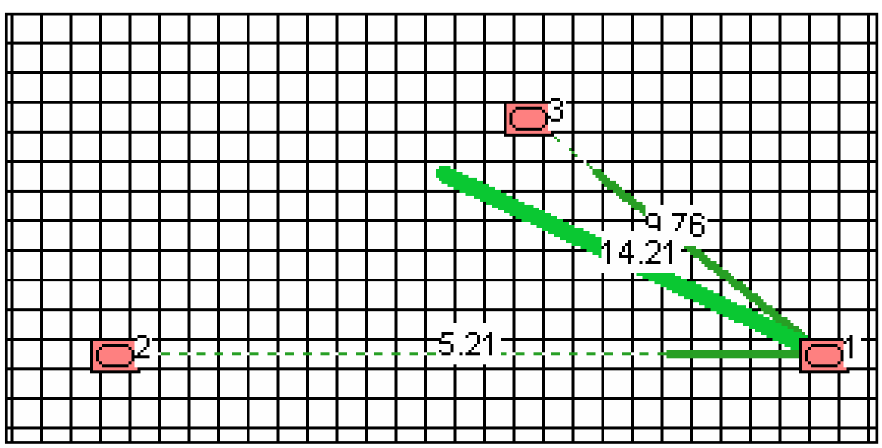


Figure 45 - Red Unit 1 is 'attracted' to the vector sum of Red Unit 2 and Red Unit 3 (displayed as thick green line). TIGER screen shot.

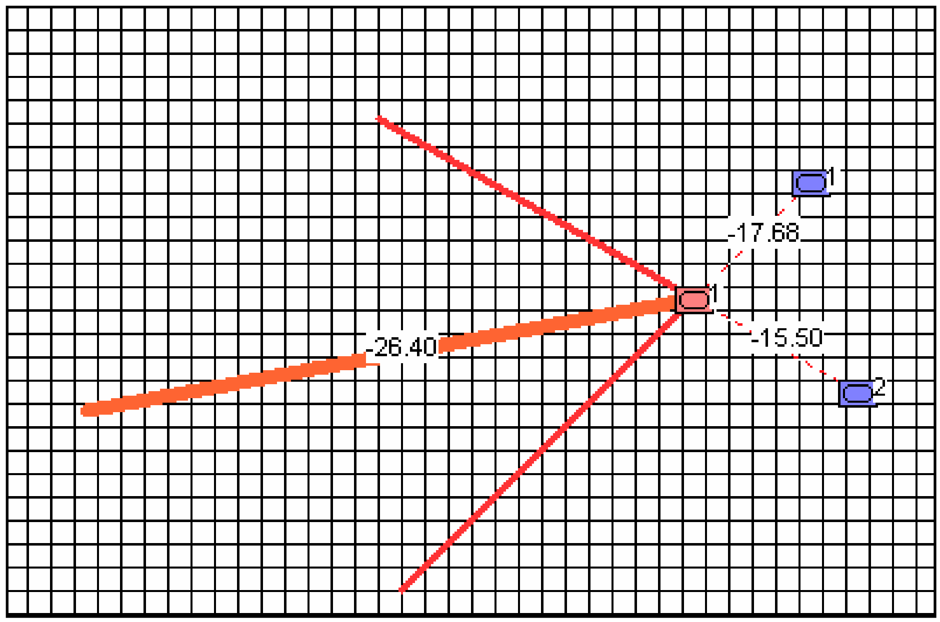


Figure 46 - Red Unit 1 'flees' along the vector sum (displayed as thick red line) of Blue Unit 1 and Blue Unit 2. Note: that *magnitude*' < $magnitude_{Blue1}$ + $magnitude_{Blue2}$. TIGER screen shot.

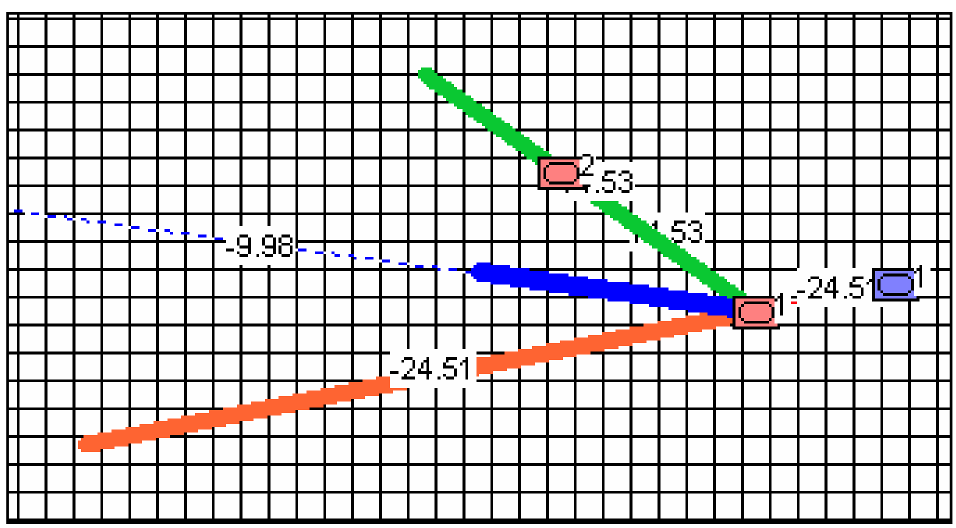


Figure 47 - Red Unit 1 'flees' from the vector of Blue Unit 1 (thick red line) and is 'pulled towards' the vector of Red Unit 2 (thick green line). The resulting vector is displayed as thick blue line. TIGER screen shot.

Appendix 2

DETERMINING LINES AND FRONTAGES

The first step of tactical calculations is determining the lines, or frontages, of the red and blue forces. Each of the offensive and defensive maneuvers has a different requirement for line and frontage analysis.

The first case that we will examine is the ‘penetration’ maneuver (see Figure 48). This maneuver requires that we first determine any gap or weak point in the enemy line. This weak point is often termed the “*schwerpunkt*”; a German word (first used in reference to Blitzkrieg maneuvers) that translates as “the point of maximum effort’ or “center of gravity.”

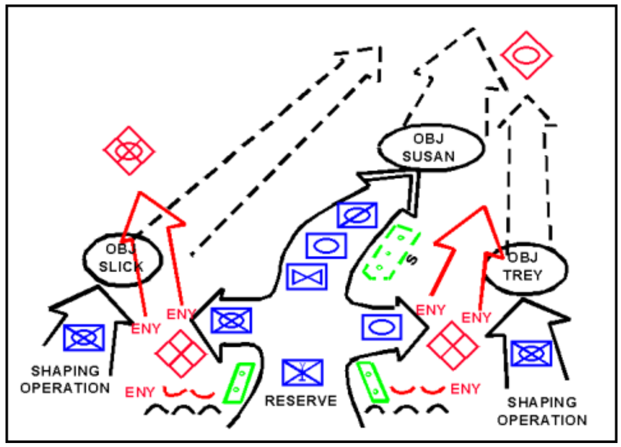


Figure 48 – The offensive ‘penetration’ maneuver. Source: U. S. Army Field Manual 3.21: Section II

In Figure 49 we have first cleared the terrain and elevation layers which eliminate all line of sight restrictions and allow us to view the units and the results of the AI calculations. We currently employ two different methods for calculating the *schwerpunkt* for the penetration maneuver: average of group locations and Acyclic MST .

The green line in Figure 49 shows the shortest weighted edge that connects Red Group 0 and Red Group 1. The *schwerpunkt* is calculated as the midpoint of this edge. Note that when we use the average weight the *schwerpunkt* is 'pulled' towards the larger Red Group 0. While this might appear as flaw, further research may prove that use of the inverse of this calculation will be the most effective implementation of this maneuver. We also intend to implement a test-bed that will analyze our different methods of calculating *schwerpunkt* for various maneuvers and determine which of the methods is optimal.

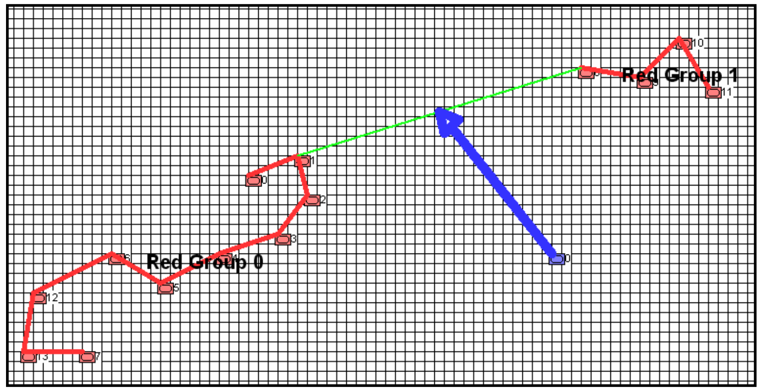


Figure 49 – *Schwerpunkt* calculated using the Minimum Spanning Tree method (TIGER screen shot). The *Schwerpunkt* is calculated as the center of the shortest edge between the two Red Groups.

For other maneuvers, such as "Envelopment" and "Turning" we wish to calculate the lines and frontages by the "Range of Influence" method. TIGER currently supports 12 distinct unit types (for both red and blue groups the distinct unit types are: armor, mechanized infantry, infantry, cavalry, and artillery, as well as a 'special' that can be used as needed). Range of Influence is related to 'influence mapping). {Sweetser 2004:439-446}

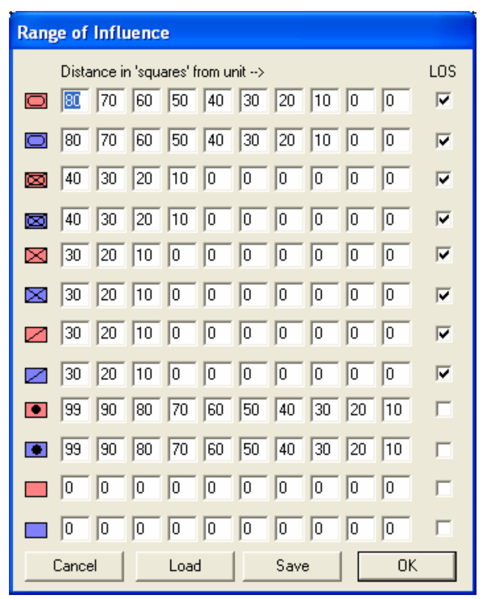


Figure 50 – Dialog box for editing Range of Influence for each unit type. Note: checking the LOS (Line of Sight) box determines if the Range of Influence is affected by intervening elevation. TIGER screen shot.

Figure 51 shows an example of groups calculated with range of influence (note: line of sight is turned off for this example).

Figure 52 displays an example of range of influence frontage calculations with line of sight active. Note: line of sight is reciprocal; the red unit must be visible to the blue unit and the range of influence is calculated on what is visible to the red unit.

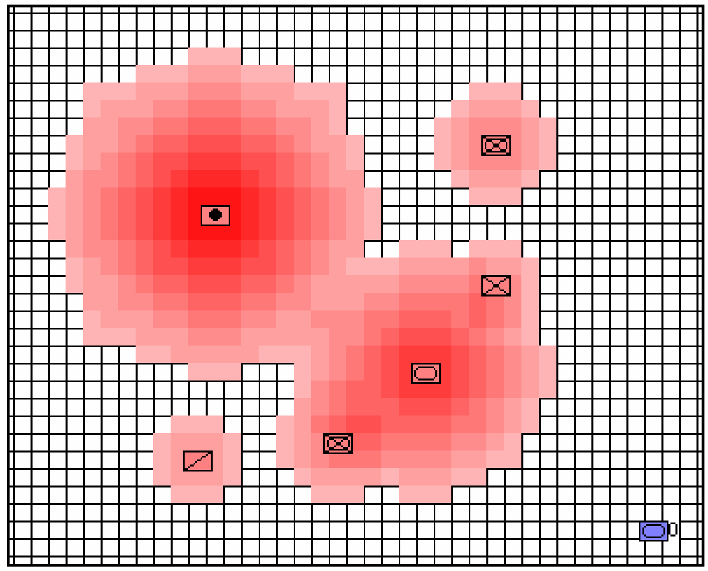


Figure 51 – Range of Influence mapping (see Figure 33 for data entry dialog). TIGER screen shot.

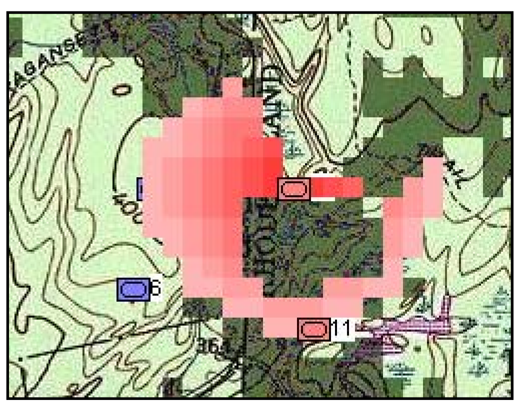


Figure 52 – Range of Influence mapping with line of sight active. TIGER screen shot.

Appendix 3

TACTICAL LINE STRUCTURES

A Tactical Line Structure is a data structure designed to store information about lines and frontages of an army. There is a Tactical Line Structure for every army in a Tactical Problem Set. The relationship between a Tactical Line Structure, armies, and units is described in Figure 53, below.

The second parameter in the CalculateLines function is a pointer to a function that is used to determine the weight of an edge between two units. The weight of the edge can be based on distance, unit types, weaponry, line of sight, or other parameters as needed.

The WorldView data structure is the implementation of the formal definition of a unit (see Appendix 5, Formal Definitions).

See Appendix 4 for an example of a Tactical Line Structure TLSdump.txt file.

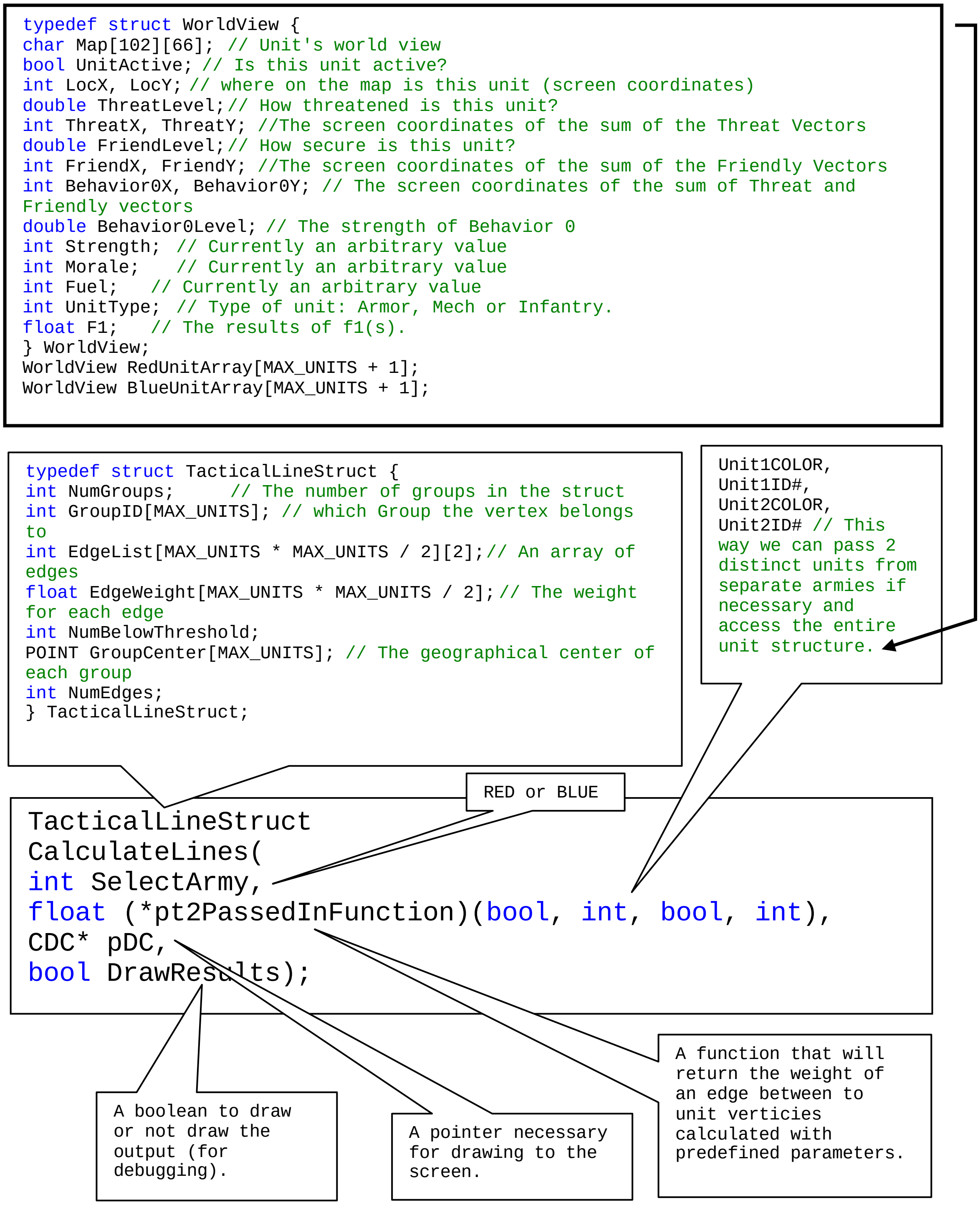
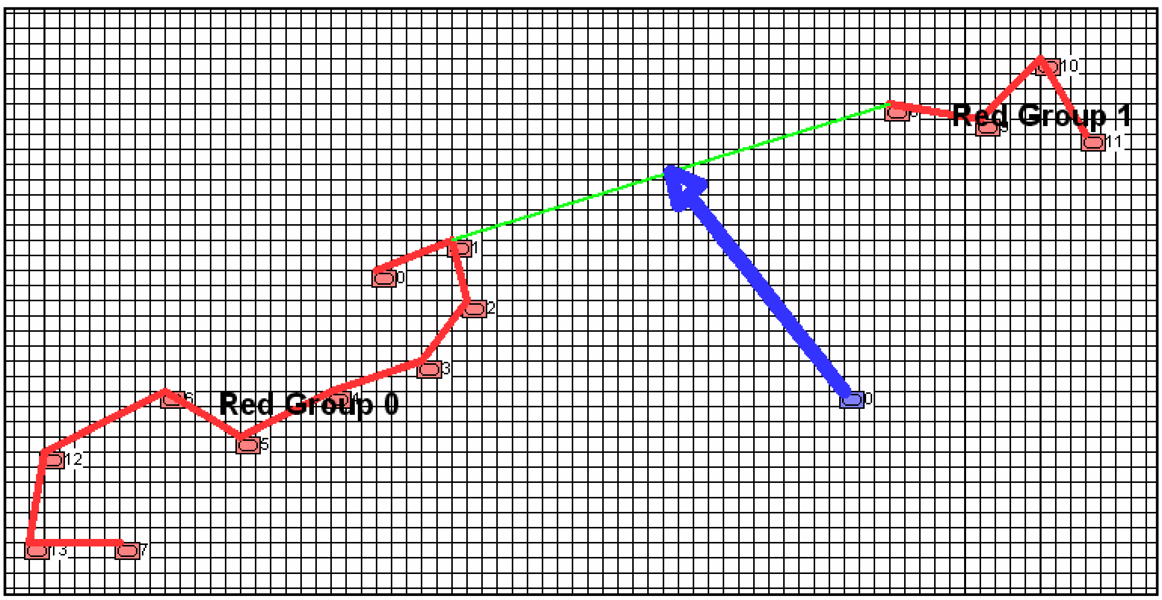


Figure 53 – The relationship between the Tactical Line Structure, armies and units.

Appendix 4

SAMPLE TACTICAL LINE STRUCTURE DUMP

Below are a sample screen shot and the resulting TLS.txt dump. Note: not all information from the Tactical Line Structure is copied to the TLS.txt file; only vertices (units), edges, edge weights, number of groups and the Group Identifier for each unit.




```
----------------------------------------
TIGER: Tactical Inference Generator v .9
Date: 02/15/07. Time: 10:53:35.
TLS Dump
----------------------------------------
NumEdges: 91
Edge  Unit ID  EdgeWeight
----------------------------------------
  0     1-2    41.231056
  1     2-3    50.000000
  2     0-1    53.851646
  3     9-10   56.568542
  4     5-6    58.309521
  5    10-11   58.309521
  6     7-13   60.000000
  7     8-9    60.827625
  8    12-13   60.827625
  9     0-2    63.245552
  10    3-4    63.245552
  11    0-3    67.082039
  12    4-5    67.082039
  13    9-11   70.710678
```

14	7-12	78.102493
15	1-3	82.462112
16	0-4	85.440041
17	6-12	89.442719
18	6-7	104.403069
19	8-10	104.403069
20	5-7	106.301460
21	2-4	108.166542
22	4-6	110.000000
23	1-4	128.062485
24	3-5	130.000000
25	5-12	130.384048
26	8-11	131.529465
27	6-13	134.536240
28	0-5	142.126709
29	5-13	156.524765
30	0-6	161.245148
31	3-6	171.172424
32	4-7	172.046509
33	2-5	174.928558
34	1-5	191.049728
35	4-12	194.164871
36	2-6	208.806137
37	1-6	214.709106
38	4-13	223.606796
39	3-7	233.238083
40	0-7	247.588364
41	0-12	250.599289
42	3-12	257.099213
43	2-7	280.178528
44	3-13	286.356415
45	0-13	292.061646
46	1-7	297.321381
47	2-12	297.321381
48	1-8	303.644531
49	1-12	304.138123
50	2-8	308.706970
51	2-13	331.209900
52	1-13	344.093018
53	3-8	353.553406
54	0-8	357.351379
55	1-9	359.026459
56	2-9	360.555115
57	3-9	403.112885
58	1-10	408.044128
59	0-9	412.310577
60	2-10	412.310577
61	4-8	415.932678
62	2-11	424.499695
63	1-11	425.793365
64	3-10	456.179779
65	0-10	461.735870
66	3-11	464.865570
67	4-9	466.154480
68	0-11	478.539459
69	5-8	483.011383
70	6-8	516.236389
71	4-10	518.941223
72	4-11	528.109863
73	5-9	533.104126
74	6-9	569.209961
75	5-10	586.003418
76	7-8	586.685608
77	5-11	594.642761
78	8-12	605.392456
79	6-10	620.322510
80	6-11	633.245605
81	7-9	635.059082

```
  82    8-13   639.531067
  83    9-12   657.875366
  84    7-10   688.839600
  85    9-13   689.420044
  86    7-11   694.622192
  87   10-12   709.365906
  88   11-12   721.248901
  89   10-13   742.495789
  90   11-13   750.266602

NumBelowThreshold: 29

-----------------------------------------
NumGroups: 2
Unit: 0.  Group: 0
Unit: 1.  Group: 0
Unit: 2.  Group: 0
Unit: 3.  Group: 0
Unit: 4.  Group: 0
Unit: 5.  Group: 0
Unit: 6.  Group: 0
Unit: 7.  Group: 0
Unit: 8.  Group: 1
Unit: 9.  Group: 1
Unit: 10. Group: 1
Unit: 11. Group: 1
Unit: 12. Group: 0
Unit: 13. Group: 0
-----------------------------------------
```

Appendix 5

FORMAL DEFINITIONS

The basic component of a simulation is the unit. A unit is a cohesive military force; frequently in a hierarchal Order of Battle Table.

Definition 2: A ***unit*** (U) is a 6-tuple (Ω, L, S, Γ, F, t) where

1. Ω: M x N is the *World View Map* of the unit which is an M x N matrix,

2. L is a tuple (x, y) that is the *Location* of the unit within the M x N matrix,

3. S is the *Strength* of the unit,

4. Γ is the *Morale* of the unit,

5. F is the *Fuel* of the unit,

6. t ⊂ T is the *Type* of the unit which is drawn from a finite set of *Types*.

■

Armies are comprised of units.

Definition 3: An ***army*** (A) is an array of ***units.***

$$A = U_{i...n}$$

■

A Tactical Problem Set is the framework that describes a military situation. A Tactical Problem Set is a 'moment frozen in time', or a state in state space, of a battle or simulation; indeed, a battle may be described as a sequential series of Tactical Problem Sets.

Definition 4: A ***Tactical Problem Set*** is a 5-tuple (Θ, E, R, B, V) where

1. Θ: M x N is the *Terrain Map* which is an M x N matrix,

2. E: M x N is the *Elevation Map* which is an M x N matrix,

3. R is the *Red Army* (A_{red}),

4. B is the *Blue Army* (A_{blue}),

5. V is a finite set called the ***victory conditions***.

■

An *attractor* is *subgoal* which can be manually placed or placed by the AI within TIGER.

Definition 5: An ***attractor*** is a triple (L, Λ_{Blue}, Λ_{Red}) where

1. L is a tuple (x, y) that is the *Location* of the attractor within the M x N matrix,

2. Λ_{Blue} is the *Value* of the *attractor* for the *Blue* army,

3. Λ_{Red} is the *Value* of the *attractor* for the *Red* army.

■

WORKS CITED

Ballard, Chris, and Ben Snively. "Toward Automated Decision Making in OneSAF Objective System: Relevant Research and a Hybrid-CBR Design Proposal," 2004.

Brooks, Rodney A. "A Robust Layered Control System For a Mobile Robot." *Technical Report: AIM-864* Technical Report: AIM-864 1985.

Burges, Christopher J. C. "A Tutorial on Support Vector Machines for Pattern Recognition." *Data Mining and Knowledge Discovery* 2 (1998): 121-67.

Chang, Chih-Chung, and Chih-Jen Lin. "LIBSVM: A Library for Support Vector Machines (Version 2,31)," 2001.

Chia, Chien Wei, and Kent E. Williams. "A Modified Naive Bayes Approach for Autonomous Learning in an Intelligent CGF."

Crawford, Chris. *On Game Design*. Indianapolis, IN: New Riders, 2003.

Department of Defense. *Dictionary of Military and Associated Terms, Joint Publication 1-02*. Washington: Department of Defense, 2007.

Department of the Army, Field Manual 3-21. *The Stryker Brigade Combat Team Infantry Battalion*. Washington: Department of the Army, 2003.

DesJardins, Marie E, et al. "A Survey of Research in Distributed, Continual Planning." *American Association for Artificial Intelligence* Winter 1999 (Winter 1999): 13-22.

Dunnigan, James F. *The Complete Wargames Handbook: How to Play, Design & Find Them*. New York: William Morrow, 1992.

---. "First Use of Computers in a Wargame?" February 2 2007. February 2. Jfdunnigan@verizon.Net.

Falkener, Edward. *Games Ancient and Oriental and How to Play Them : Being the Games of the Ancient Egyptians, the Hiera Gramme of the Greeks, the Ludus Latrunculorum of the Romans and the Oriental Games of Chess*. New York: Dover, 1931.

Griess, Thomas E. *Atlas for the American Civil War*. Wayne, NJ: Avery Publishing Group, 1986.

---. *Atlas for the Arab-Israeli Wars, the Chinese Civil War and the Korean War*. Wayne, NJ: Avery Publishing Group, 1986.

---. *Atlas for the Second World War (Europe and the Mediterranea)*. Wayne, NJ: Avery Publishing Group, 1986.

---. *Atlas for the Wars of Napoleon*. Wayne, NJ: Avery Publishing Group, 1986.

Hsu, Chih-Weih, Chih-Chung Change, and Chih-Jen Lin. “A Practical Guide to Support Vector Classification,” 2001.

Lockheed Martin Corporation. *OneSAF Testbed Baseline Assessment*. 1998.

Longstreet, James. *From Manassas to Appomattox*. New York: Konecky & Konecky, 1992.

Osborne, Martin J. *An Introduction to Game Theory*. New York: Oxford University Press, 2004.

Penner, Robin R., and Erik S. Steinmetz. “JointAdvisor: An Intelligence Analysis Agent,” 2001.

Perla, Peter P. *The Art of Wargaming*. Annapolis, MD: Naval Institute Press, 1990.

---. “First Use of Computers in a Wargame?” February 1 2007 <email>. Perla@cna.Org.

Robel, Michael K. “First Use of Computers in a Wargame?” February 1 2007 <email>. Mkrobel1@cs.Com.

Ryan, Peter, Peter Clark, and Lucien Zalcman. “Distributed Interactive Simulation for RAN Training.”

Sacerdoti, Earl. D. “The Nonlinear Nature of Plans.” *IJCAI* (1975).

Sidran, David E. “A Brief History of Computers & Wargaming,” 1993.

Sweetser, P. *AI Programming Wisdom 2*. Ed. Steve Rabin. Vol. 2. Hingham, MA: Charles River Media, Inc., 2004. Strategic Decision-Making with Neural Networks and Influence Maps.

Sweetser, P, and J. Wiles. “Combining Influence Maps and Cellular Automata for Reactive Game Agents.” Paper presented at the 6th International Conference on Intelligent Data Engineering and Automated Learning, 2005.

Tate, Austin. “The Less Obvious Side of NONLIN.” *(Unpublished)* (1983).

Tate, Austin, Jeff Dalton, and John Levine. “O-Plan: A Web-Based AI Planning Agent.” *American Association for Artificial Intelligence* (2000).

Turing, Allan M. “Computing Machinery and Intelligence.” *Mind* LIX.No. 236 (October 1950): 433.

Woodcock, Steven. “Recognizing Strategic Dispositions: Engaging the Enemy.” *AI Programming Wisdom*. Ed. Steve Rabin. Hingham, MA: Charles River Media, Inc., 2002. 221-32.

Yu, Leigh. “A Bayesian Network Representation of a Submarine Commander.”