# Models of Defeat

Gary King Brent Heeringa David Westbrook Joe Catalano and Paul Cohen

Department of Computer Science
University of Massachusetts, Amherst
140 Governor's Drive
Amherst, MA 01003
{gwking,heeringa,westy,catalano,cohen}@cs.umass.edu

**Abstract.** Capture the Flag is a wargaming environment that includes intelligent, autonomous adversaries. In the past, the planner controlling the adversaries focused on two goals: occupying objectives and attrition. However, attrition is actually a means to an end, defeat of one's enemies, not an end in itself, and not necessarily desirable. Similarly, coalition operations plan for defeat by applying force for political and psychological effect. For Capture the Flag to plan for these effects, it needs a model of defeat. We model the "capacity for conflict" as a leaky bucket: when a unit's bucket is full, it has no more capacity for conflict and it capitulates. Flow into and out of the bucket is modulated by several factors including attrition and heroism. The model is inherently dynamical, so it exhibits the time-dependent behaviors one observes in real conflicts; for example, identical attacks will have different effects on capitulation as a function of their timing.

## 1 Introduction

Coalition operations are increasingly effects-based, which means they apply force only as necessary to achieve political and psychological effects. Actions that have relatively small effects in terms of conventional target-based or attrition-based planning can have large political and psychological effects, not only on adversaries but also on coalition members. AI planning technology has not kept up with the requirements of effects-based coalition planning, in part because we lack *models* of psychological and political effects. It is relatively easy to model the effects of attrition on conventional units — they get smaller, less mobile, less lethal, and so on — but what about their psychological state, their will to fight, their morale? Where are the models to predict the catastrophic collapse of the Iraqi regular units, or the differences between Taliban fighters from Saudi Arabia and those from Afghanistan?

This paper reports on our efforts to add models of defeat mechanisms to the Capture the Flag wargaming system. Defeat mechanisms are strategies that achieve capitulation. While attrition may achieve defeat, it is not necessarily the fastest or most desirable course of action. Although most planners are attrition-based, intelligent wargaming environments need agents that use defeat mechanisms to plan for the effects of their actions. For example, a smart agent might notice that an opposing unit has separated from its supply line and is ripe for an attack. It might also notice that attacking from a nearby forest is better than other routes because it will surprise the foe. While filling planners with rules like *always initiate attacks from hidden terrain* is possible, it is not necessarily desirable. Instead, we want agents that plan for effects: attacking an isolated unit from the forest is good because it is more easily defeated. That is, the effects of isolation and surprise make the foe more susceptible to defeat because its capacity for conflict is reduced.

If an agent is to plan for defeat it requires a model. This paper presents one such model that uses a metaphoric leaky bucket to represent an agent's capacity for battle. The bucket has inputs, outputs, and effects. While conceptually simple, the leaky bucket model paired with the Capture the Flag environment is flexible enough to account for many non-linear effects of battle. For example, differences in unit type, impact of friendly and opposing forces, and soft factors such as morale can all be represented with the leaky bucket and contribute, non-linearly at times, to defeat.

In the next section we review previous work in modeling defeat and discuss how our model differs from current research. Next we discuss the Capture the Flag wargaming environment and how our model naturally complements the simulator and planner. We follow this with details of the leaky bucket, specifically the mathematics of input and output flows and how they affect an agent's physical attributes. We demonstrate the flexibility and effectiveness of the model through a number of experiments and conclude with a discussion of future work.

## 2 Background

Historically, defeat models use hardcoded breakpoints of casualties to determine surrender or posture changes (Dupuy, 1990). Most researchers agree that such models are inaccurate and ill-suited for simulation. In particular, Dorothy Clark found that the breakpoints of casualty ratios in historical data fell uniformly between 10 and 70 percent (1954). She concludes that factors such as breakdown in leadership, support, reinforcement, and communication affect capitulation more than attrition. It was not until recently though, that such models were addressed for the purpose of simulation.

Janice Fain in (1990) provides two of the first non-attrition based breakpoint models for determining posture changes in computer simulation. Both models are essentially flow charts of conditional statements, but one flows with respect to time, the other by event. The variables in each condition are taken from historical data, interviews with veterans, and facts from the literature. The variable thresholds are calculated exclusively from battle data. This tends to model the historical

data well, but may lead to overfitting. Fain's methodology also set the trend for future research: identify factors that influence defeat and model them directly.

In this spirit, a wealth of related literature exists both in the decision-making and human behavior modeling communities. For example Hudlicka and Billingsley (1999) develop a cognitive architecture for modeling individual characteristics such as personality and affective factors; McKenzie et al. (2001) investigate human personality models in decision-making; Gillis and Hursh (1999) integrate human performance models into simulation; and Angus and Heslegrave (1985) discuss cognitive abilities during command and control exercises in the event of sleep loss.

Recent work that deals directly with models of defeats include Alan Zimm's casual model of warfare (1999), and Brown and May's work in casting defeat as a breakdown in organizational structure within a complex adaptive system (2000). Zimm's work primarily identifies "stress factors" and their "cause and effect relationships" with a unit's behavior and decision-making skills. In contrast, Brown and May take a more biological slant on capitulation. When a unit can no longer adapt to the battle and its environment, it is primed for defeat. This argument is compelling and probably deserves more attention.

With the exception of Brown and May's research, most of the current and related work in defeat models deals with first-level characteristics of individuals. In Capture the Flag, this level of granularity is inappropriate since the agents are spring and blob masses and interact in an abstract world. In the next section we describes the Capture the Flag wargaming environment and how the bucket model works within and complements the system.

## 3 Capture the Flag

Capture the Flag is based in the Abstract Force Simulator (AFS) (Atkin et al., 2001; Atkin et al., 1999). AFS is a simulator of processes that operates with a small set of physical features including mass, velocity, friction, radius, attack strength and so on. The agents in AFS are abstract units called blobs; a blob can be an army, a soldier, or a political entity. Every blob has a small set of primitive actions it can perform: PRIMITIVE-MOVE, APPLY-FORCE (push), and CHANGE-SHAPE. All other actions are built from these. Blobs are modeled as a set of balls connected by springs where balls are point masses that can exert a force at some distance from their center. The ball and spring model means that blobs are amoeba-like: they can assume almost any two dimensional shape without holes.

We create simulations by changing the physics of AFS–how collisions affect mass and velocity, how terrain surfaces affect friction and so on. By tuning AFS, we have used it to simulate billiard balls, robots moving from room to room, rats scurrying about on a network of streets, and military battalions in division level combat.

AFS is tick-based, but the ticks are small enough to accurately model the physical interactions between blobs. Although blobs themselves move continuously in 2D space, for reasons of efficiency, the properties of this space, such as terrain attributes, are represented as a discrete grid of rectangular cells. Such a grid of cells is also used internally to bin spatially proximal blobs, making the time complexity of collision detection and blob sensor modeling no greater than linear in terms of the number of blobs in the simulator. AFS was designed from the outset to be able to simulate large numbers (on the order of hundreds or thousands) of blobs.

The physics of the simulation are presently defined by the following parameters:

Blob-specific parameters:

- shape
- density
- viscosity and elasticity: determine how blobs interact
- mass: the blob's ability to apply force
- position and velocity
- acceleration
- friction on different surfaces
- strength coefficient: a multiplier on mass to compute the force a blob can apply
- resilience coefficient: determines how much mass a blob loses when subjected to outside force

Global parameters:

- the different types of blobs present in the simulation (such as blobs that need sustenance or blobs than can apply force at a distance)
- the damage model: how blobs affect each others' masses by moving through each other or applying force
- sensor model: what information blobs can collect

AFS is an abstract simulator; blobs are abstract entities that may or may not have internal structure. AFS allows us to express a blob's internal structure by composing it from smaller blobs, much like an army is composed of smaller organizational units and ultimately individual soldiers. Because a blob is completely characterized by its physical attributes at every level of abstraction, we can ignore its internal structure while simulating if we choose to. Armies can move and apply force just as individual soldiers do. The physics of armies is different than the physics of soldiers, and the time and space scales are different, but the main idea behind AFS is that we can simulate at the "army" level if we so desire—if we believe it is unnecessary or inefficient to simulate in more detail.

In a similar fashion, we use abstract notions like *mass*, *strength* and *resilience* as stand-ins for the vast variety of actual unit attributes: weapon type, training, ammunition levels, supply lines, sickness, and so on. The mass of a blob agglomerates all of these and its strength and resilience account for the broad strokes of situation dependent factors. This

loss of detail allows Capture the Flag simulations to be built and run in minutes rather than days. We can quickly assess multiple COA's in Monte Carlo trials and use our understanding for refinements in planning and strategy. Our simulation could be made much more detailed, but doing so runs the risk of arbitrary parameter choices and of pretending knowledge about what is best captured as noise and variance.

## 4 The Leaky Bucket Model

Innumerable factors influence whether or not an agent will cease to function. In Capture the Flag, we combine all of these factors into one abstract quantity we call fatigue. The fatigue of an agent rises and falls depending on its activities and interactions. Fatigue also alters these activities and interactions because an agent's fatigue changes its effectiveness. For example, as an agent's fatigue rises, it becomes less able to exert force and to protect itself, it moves more slowly, processes information less accurately, and so on. Finally, the agent has a breaking point. When an agent's fatigue becomes higher than this preset amount, the agent ceases to function.[1]

Using $\mathcal{F}_t$ and $\mathcal{E}_t$ to represent the fatigue and effectiveness (respectively) of an agent at time $t$, the following two general equations relate fatigue and effectiveness.

$$\begin{aligned} \mathcal{F}_t &= \mathcal{F}_{t-1} \\ &\quad +f(\mathcal{F}_{t-1}) \quad \text{Loss} \\ &\quad -g(\mathcal{F}_{t-1}) \quad \text{Recovery} \end{aligned} \tag{1}$$

$$\mathcal{E}_{t,i} = h_i(\mathcal{F}_{t-1}) \tag{2}$$

The new fatigue of an agent depends on its previous fatigue and two functions $f$ and $g$ which increase and decrease it. The effectiveness of an agent depends on another function $h$. In Capture the Flag, agent effectiveness is modeled as a multiplier on its strength, resilience, friction, turn rate, enemy intelligence abilities, sighting ability and so on. We call these altered agent properties the *effects* of fatigue. We subscript $\mathcal{E}$ and $h$ to indicate that the change occurs for each altered agent property $\mathcal{P}$.

By varying the functions $f$, $g$ and $h$, this model can become arbitrarily complex. We have chosen to keep these functions simple initially and to only add complexity when it seems necessary. We use the following functions:

$$f = \mathcal{M}_{self,t} + \frac{\mathcal{M}_{self,t}}{\mathcal{M}_{attacker,t}+\mathcal{M}_{self,t}} \quad \text{Differential mass loss} \tag{3}$$

$$g = \mathcal{R} \quad \text{a constant recovery factor} \tag{4}$$

$$h_i = \mathcal{P}_i(1 \pm \kappa\frac{\mathcal{F}_t}{\mathcal{B}}) \quad \text{for each effect} \tag{5}$$

Where we use the following notation:

| | |
|---|---|
| $\mathcal{B}$ | Breaking point |
| $\mathcal{P}_i$ | Agent property effected by fatigue |
| $\mathcal{M}_{self,t}$ | Agent's mass lost at time $t$ |
| $\mathcal{M}_{attacker,t}$ | Combined attacker's mass lost at time $t$ |
| $\kappa$ | The maximum percentage change in effectiveness due to fatigue. |

We use the $\pm$ notation in equation 5 because some properties decrease as fatigue increases (e.g., strength and resilience multipliers) while others increase along with fatigue (e.g., friction). Each $h_i$ will use the appropriate operation. We also make the following simplifying assumptions:

- Although the actual initial breaking point of an agent depends on its training, motivation and other intrinsic factors, we model it based solely on unit type.
- Each agent has a constant recovery rate that reduces its fatigue over time.
- The fatigue of an agent increases when it loses more mass that the units attacking it lose (i.e., the increase is based on the (perceived) differential mass loss).
- Rather than modeling each effect separately, we use the same percentage change in effectiveness for all of them.

[1] In the current implementation, agent's that have broken are removed from the game. We are considering providing agents with the ability to reconstitute and also with multiple breakpoints.

To summarize our model: an agent's fatigue rises when it is damaged and especially when its enemies damage it more than it damages them. The fatigue also has a natural constant recovery rate. The fatigue has a linear effect on the effectiveness of the agent where effectiveness is modeled by scaling the agent's key properties away from their nominal values. Though it is not explicit, this model is non-linear because as the fatigue rises, the effectiveness falls and as the effectiveness falls, the agent is liable to take more damage (and dole out less) which will cause the fatigue to rise more quickly.

One of the putative advantages of our model is that it has few parameters and all of them are reasonably intuitive:

| | |
|---|---|
| $\mathcal{B}$ | Breaking point or bucket size |
| $\kappa$ | The maximum percentage change in effectiveness due to fatigue. |
| $\mathcal{R}$ | a constant recovery factor |

But these alone allow us to model different training levels (via increased bucket size or smaller $\kappa$); resilience to stress (by increasing $R$) and so on. By modifying $f$, $g$, and $h$ we can complicate the model as necessary.

## 5 Experimental Results

As both Capture the Flag and our leaky bucket model operate in the abstract physics of AFS, it makes little sense to ask for quantitative results. Instead, we validate our model by seeing how well it matches the qualitative interactions of real military conflict. Any reasonably complex model can be tuned to fit almost any desired outcome. Our goal is to see if our simple model provides the right sort of behaviors without endless tuning. This section presents results from three different simulations

**Two evenly matched blobs** We ran 300-trials varying two independent variables: 1. Defeat Model: this could be on, off or on for only one blob; 2. Bucket size: this could be large or small. In each case, we randomly varied the initial mass and positions of the blobs.

**A small blob against a much larger opponent** We ran 50-trials in each of five conditions by setting the initial bucket level of the larger blob to 0-%, 30-%, 50-%, 70-% or 90-% of it maximum size.

**Two blobs attacking a single, much larger blob** We ran a total of 600-trials varying the total effect of the model ($\kappa$ =10-% or $\kappa$ =90-%) and the number of ticks between the attacks of the two blobs (0-, 15-, 30-, 45-, 60-, and 90-ticks). We also randomized the initial mass of each of the blobs. We did not randomize the positions because doing so added too much additional variance to the delay between the attacks.

In each simulation, we investigate if our model produces reasonable results.

### 5.1 Two evenly matched blobs

We might expect battles to last longer if blobs become less effective as they become fatigued–think of two drunken and weary boxers. On the other hand, if blobs can break and surrender, we might think that battles should end more quickly. We can observe both of these effects in this simulation. When the model is turned on, smaller bucket sizes lead to shorter battles (47-ticks as compared to 60-ticks for the larger bucket size). On the other hand, battles between blobs with high breaking points actually last longer (60-ticks as compared to 57-ticks) than the same blobs with no defeat model. Note that these battles may *last* longer but they actually do *less* total damage. As expected, battles with the defeat model turned on always produce less overall attrition that those with the model turned off.

### 5.2 A small blob against a single, much larger blob

If fatigue is not a factor, a small blob can never defeat a larger enemy in a head on assault. As the larger blob becomes fatigued, however, we would expect that it will suffer more damage and possibly even reach its breaking point. Furthermore, we would expect that the smaller blob would suffer correspondingly less damage. This simulation provides qualitative evidence of exactly these effects.

### 5.3 Two red blobs attacking a single, much larger blue one

All else being equal, it is always better to coordinate attacks. Adding fatigue to the simulation should greatly exacerbate the problems of uncoordinated attacks because blobs recover somewhat between attacks (at a rate determined by the outflow constant $\mathcal{R}$). Our simulations show how a coordinated attack succeeds where an uncoordinated one cannot and furthermore show significant differences when the blob effects from the defeat model are turned up high. Figure 1 shows the result of a coordinated attack. The $x$-axis shows time (in ticks) and the $y$-axis shows how full the blue blob's bucket is as a percentage of its total size. The stars on the graph show at what ticks the two red attacks occurred. As you can see, each attack causes an inflection in the graph. Because the attacks are coordinated, the blue blob has no time to recover and is overwhelmed. Figure 2 paints a completely different picture. The axes in this graph are the same but here the two red attacks are uncoordinated. The blue blob is able to defeat the first red blob and has time to recover before the second blob attacks. This recovery time allows it to defeat the second attacker and win the day.

In sharp contrast, figure 3 shows the difference in total mass lost when model effects are high and low. When fatigue effect is high, the blue blob cannot recoup its losses even with equal recovery time.

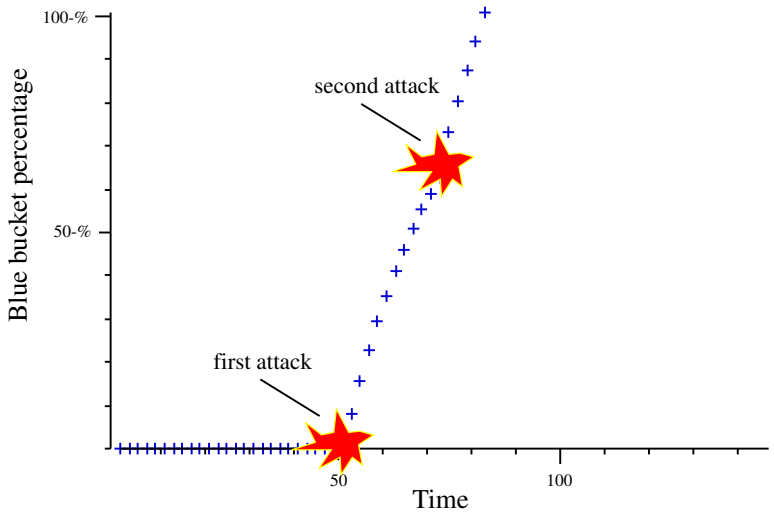


Figure 1: Blue bucket level over time, coordinated attack

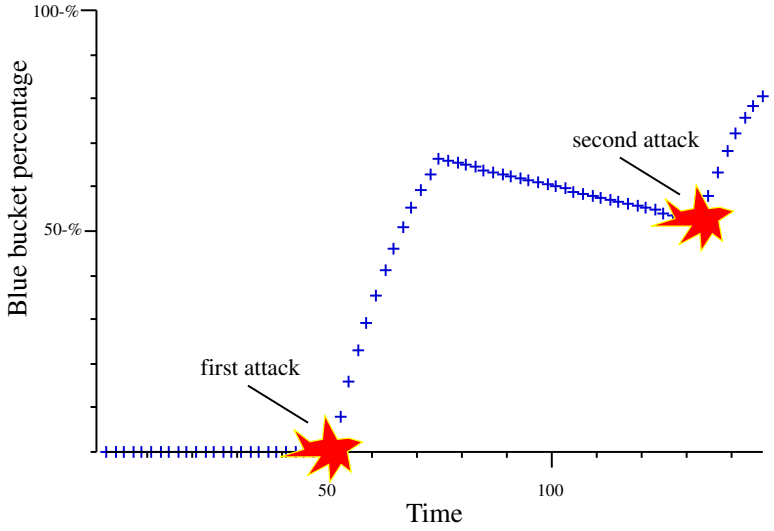


Figure 2: Blue bucket level over time, uncoordinated attack

## 6 Future Work

Our current model provides a simple, parameterized defeat model implemented within the Capture the Flag wargaming simulation. Our qualitative experiments show that the leaky bucket matches our expectations and increases the fidelity and range of Capture the Flag. There remains, however, much work to be done. For example, there are several plausible additions we can make to the individual agent model. Furthermore, although the Leaky Bucket model extends the behavior repertoire of single agents within the simulation, it does not capture the interactions between agents. Finally, we need to complete the circle and use our model to create plans which lead to capitulation by their effects rather than by brute force and attrition.

### 6.1 Model Extensions

The current model is deterministic whereas real battles are always characterized by the unexpected bravery or cowardice of individuals. We can capture the flavor of these events by adding a stochastic element to the bucket inflow and outflow functions ($f$ and $g$). This would occasionally cause large decreases or increases in an agent's bucket level leading to renewed vigor or sudden defeat.

The current model also seems impoverished in its overreliance on blob combat as the only means of bucket level increase. We intend to investigate isolation, perceived vulnerability and terrain unfamiliarity as possible new sources of inflow. Some of these relate to the group dynamics of agents operating together to achieve their goals.

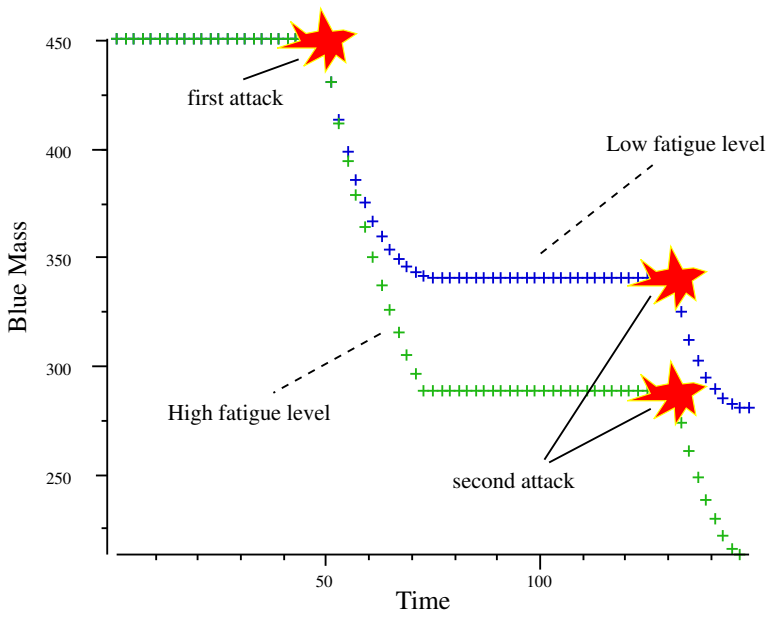


Figure 3: Blue mass level over time in uncoordinated attack comparing high and low effects of fatigue

## 6.2 Group Dynamics

The fatigue and morale of agents cannot really be viewed in a vacuum. Agents interact to uplift and poison one another; they respond to events all over the battlefield and in the world beyond; and they respond differently depending on their current situation and location. We will model all of these interactions by extending Capture the Flag with a layered network model of interconnections modeling Command and Communication, Supply, Infrastructure and so on (Cohen et al., 1996). Events will pass over this network and act as inflows and outflows on each agent's bucket.

## 7 Conclusion

Models of defeat are an integral component of intelligent wargaming environments for two reasons. First, models of defeat make simulation more realistic and agent behavior more accurate. Second, they provide a means for agents to execute defeat mechanisms – courses of action that achieve capitulation in military engagements. We presented a conceptually simple leaky bucket model that interacts with our Capture the Flag wargaming environment to capture many non-linear effects of defeat mechanisms. Qualitatively, our model behaves realistically and reasonably under a variety of different scenarios. In particular, we showed that the model is sensitive to the timing of attacks: coordinated attacks succeed whereas uncoordinated attacks fail. In the future we will experiment with agents that plan for the effects of defeat mechanisms. Such planning combined with our leaky bucket defeat model should result in a robust and realistic wargaming environment.

## Acknowledgements

This research is supported by the Defense Advanced Research Projects Agency (DARPA) and Air Force Research Labs (AFRL) under contracts 30602-99-C-0061 and F30602-01-1-0589. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright notation hereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements either expressed or implied, of the DARPA, AFRL or the U.S. Government. Many thanks to General Charlie Otstott for his valuable comments and observations.

## References

Angus, R. G., & Heslegrave, R. J. (1985). Effects of sleep loss on sustained congnitive performance during a command and control simulation. *Behavior Research Methods, Instruments, and Computers, 17*, 55–67.

Atkin, M. S., King, G., Westbrook, D., Heeringa, B., Hannon, A., & Cohen, P. R. (2001). Hierarchical agent control: A framework for defining agent behavior. *Proceedings of the Fifth International Conference on Autonomous Agents* (pp. 425–432). ACM Press.

Atkin, M. S., Westbrook, D. L., & Cohen, P. R. (1999). Capture the Flag: Military simulation meets computer games. *Proceedings of AAAI Spring Symposium Series on AI and Computer Games* (pp. 1–5).

Brown, M., & May, A. (2000). Defeat mechanisms: Military organizations as complex, adaptive, nonlinear systems. Prepared for the Office of Net Assessment, Office of the Secreatary of Defense, under Contract No. DASW01-95-D-0060.

Clark, D. K. (1954). *Casualties as a measure of the loss of combat effectiveness of an infantry battalion*. Washington D.C.: Operations Research Officer.

Cohen, P. R., Anderson, S. D., & Westbrook, D. L. (1996). Simulation for ARPI and the air campaign simulator. *Advanced Planning Technology* (pp. 113–118). Menlo Park, CA: AAAI Press.

Dupuy, T. N. (1990). *Understanding defeat*. New York, New York: Paragon House.

Fain, J. B. (1990). Simulating defeat with computers: Forced changes of combat posture. *Understanding Defeat* (pp. 261–290).

Gillis, P. D., & Hursh, S. R. (1999). Using behavior moderators to influence cgf command entity effectiveness and performance. *8th Conference on Computer Generated Forces and Behavioral Representation*.

Hudlicka, E., & Billingsley, J. (1999). Representing behavior moderators in military human performace models. *8th Conference on Computer Generated Forces and Behavioral Representation*.

McKenzie, F. D., Catanzaro, J., & Petty, M. D. . (2001). A personality-based command decision-maker: Results and recommendations. *10th Conference on Computer Generated Forces and Behavioral Representation*.

Zimm, A. D. (1999). Modeling maneuver warfare: Incorporating human factors and decisionmaking in combat modeling. *Proceedings of the 8th Conference on Computer Generated Forces and Behavioral Representation*.