Analysis and Training — Why Different?

Joseph P. Provenzano[1]
Project Manager, Corps Battle Simulation (CBS)

Robert G. Chamberlain[1]
Principal, CBS Modeler

Timothy Metivier[2]
Combat Developer, U. S. Army

Presented[3] at the
71st Military Operations Research Society Symposium
Marine Corps Base, Quantico, VA
10-12 June 2003

ABSTRACT (UNCLASSIFIED)

It is tempting to seek universal combat simulations, optimized for both analysis and training. Such a quest would be quixotic, however, because the two environments place conflicting requirements.

Analytical simulations must give accurate assessments of differences between carefully defined alternatives in a stable environment. Training simulations must reliably reward good decisions and punish bad ones in a chaotic environment. They are as different as the two sides of a coin. Optimization for one purpose precludes optimization for the other.

Issues discussed include: Dynamic play, dynamic data modification, continuous credibility, reliability and recovery (checkpoint/modify/restart), user interfaces, behavior of subordinates, COBRA, OPFOR, gamesmanship, magic, artificial intelligence, dynamic task organization, scaling, C4I interfaces, computer performance, and world-wide turnkey fielding.

---

[1] Jet Propulsion Laboratory, California Institute of Technology, 4800 Oak Grove Drive, Pasadena, CA 91109-8099.

[2] National Simulation Center, Ft. Leavenworth, KS 66027-7305.

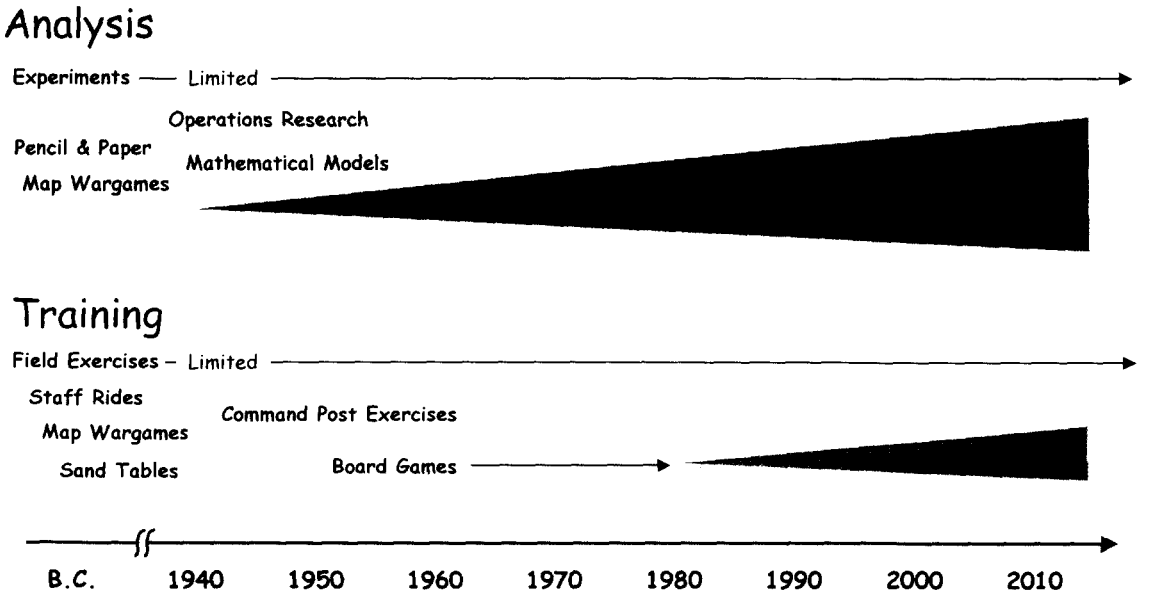
[3] The research described in this publication was carried out by the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.

Introduction

All combat simulations, whether used for analysis or for training, have much functionality in common: attrition, movement, sensing, and so on. Thus, it may be surprising to note how different the two kinds of simulations must be.

Analytical simulations are used to assess differences in effectiveness between alternative designs for equipment or tactics,[4] because combat is far too complex for closed form solutions to exist. Analysts use simulations to explore the parameters of the design space, producing statistically repeatable, accurate comparisons of the effects of small changes even when the environment is stochastic. In this context, simulation is the very powerful last resort of the competent.

Training simulations are used to provide an environment in which a training audience is exposed to the consequences of their decisions, whether good or bad. Training simulations must produce credible results every time, even though the environment is chaotic. Trainers use simulations to provide experience consistent with doctrine — that is, with the lessons the trainers believe should be taught. In this context, simulation is the only option there is.[5]



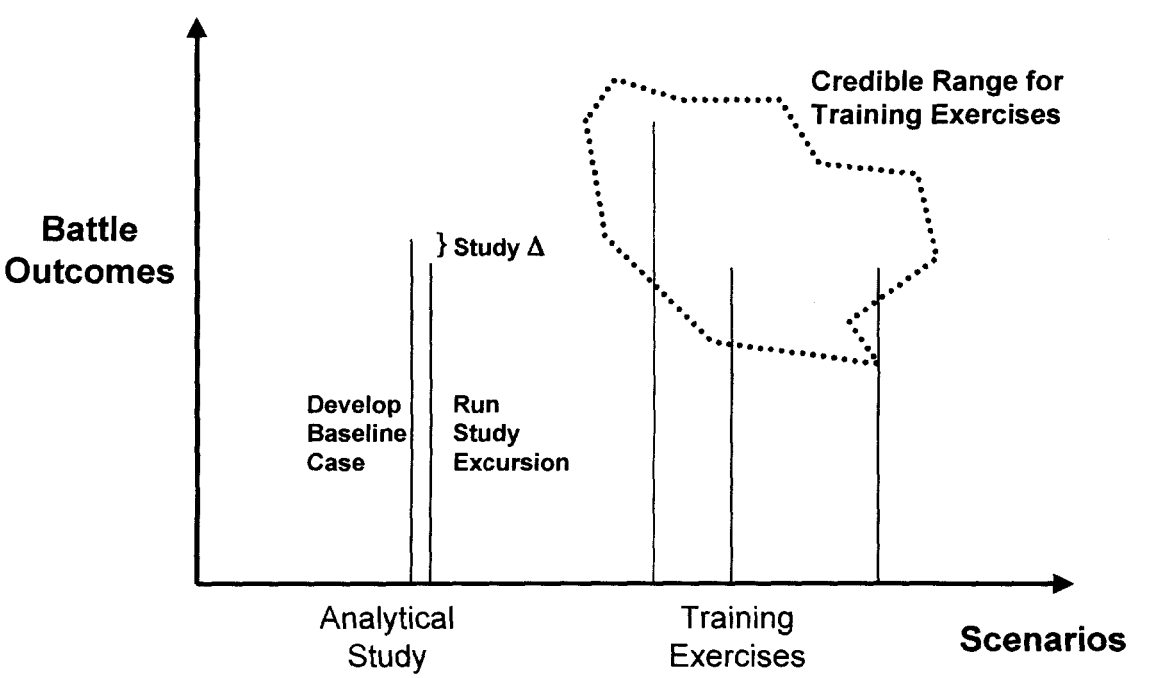
[4] Analyses may address other issues, such as how much materiel is consumed, what TO&E works best, etc.

[5] Even wartime experience itself could provide negative training, as either a foolish opponent or a brilliant subordinate could "teach" that a bad decision leads to a good result (or *vice versa*). Doctrine is based on extensive experience of senior leaders regarding the likely results of expected performance by opponents and subordinates. In a training exercise, only members of the training audience — not opponents and not virtual subordinates — should be brilliant or foolish.

Dynamic Play — Stochastic Uncertainty vs Chaos

Analytical modelers are generally interested in quantitative, rather than qualitative, differences in outcomes. For analyses, it is normal to tweak and calibrate the simulation program during the verification and validation process until it produces the expected result for a known baseline case. Analyses are then performed by changing one or more parameters while staying within the boundaries of validated results. In the most sophisticated analyses, parameters with uncertain values are allowed to vary stochastically according to known probability distributions and the statistics of selected outcome metrics are gathered. Alternatives are compared by testing statistical hypotheses.

Training, on the other hand, requires the dynamic interplay of action, reaction, and counteraction between two or more players with conflicting interests. Training audience players must be allowed to make bad decisions. The resultant outcome space is huge and chaotic. The consequences of decisions must have the correct sign[6] and be of about the right magnitude, but high accuracy is much less important than it is for analysis. Nevertheless, even unimportant details must be qualitatively credible for the important lessons to be convincing.



[6] Bad decisions should not be rewarded and good decisions should not be punished.

Dynamic Data Modification — Continuous Credibility

Unlike analytical models, training simulations cannot be revised and rerun until the results are satisfactory. Instead, it must be possible to make adjustments during the exercise. Insofar as possible, the behavior of models within the simulation must be controlled by adjustable parameters and, of course, data. Then, as can be done through the executive in CBS, the values of those parameters can be changed without even pausing the simulation.

Reliability and Recovery (Checkpoint / Modify / Restart)[7]

Reliability and recovery are at the heart of a training simulation since all the people are pulled together for a non-stop experience that spans four or five days — or longer. Even if the simulation were to contain no errors and never go down, it is sometimes (rarely) desirable in training to turn the clock back and let the training audience avoid making a tactical mistake (or, perhaps even a tactical *coup de grâce*!) that makes it impossible to achieve the training objectives that were planned for the exercise.

Even if the simulation has no erroneous code, database errors can cause crashes. In an analytical simulation, the data can be fixed and the program rerun. In a training simulation, training would be severely compromised if several days of operations had to be repeated due to *any* kind of "computer error".

No computer program containing half a million lines of evolving code is so robust that it will never go down. Unless the code can be revised before coming up from a checkpoint, the warfighter might have to be restarted from the beginning after fixing the code — clearly an untenable situation. A recovery capability provides more value to a system than an impossible quest for error-free code.

Other reasons for code modification include memory leaks, performance problems, functional bugs, and functionality that is simply unacceptable to the users, no matter how well designed and coded. Short of disabling entire areas of functionality, such situations are virtually impossible to address without the ability to modify code. Moreover, one cannot count on identifying every possible crash, memory leak, performance bottleneck, or functional issue prior to fielding the software.

If the code can be fixed, the exercise can proceed with full functionality, but the length of the delay depends heavily on how long it takes to repair the code. Provided it is not necessary to edit the input order stream, the delay can be shorter than when it is caused by database errors even though full functionality would be restored. If a full compilation were required, however, the delay would likely be prohibitive. Hence, code modification must be possible, building new libraries of executable code, without full recompilation. The baseline code can then be repaired after the exercise is over.

[7] This section is based on an unpublished work by Jay Braun.

User Interfaces

Analytical simulations interface only with the analyst, who is usually also the developer. Training simulations, on the other hand, have four kinds of dynamic users: the training audience (with its organic ABCS or other C4ISR devices), other software systems, simulation control, and exercise control.

The most demanding user interfaces support the training audience. User workstations provide situational awareness — hence, *perceived truth*, rather than *ground truth* — to role players. User workstations supply SITREPs, Battle Damage Reports, and so on. They may be of several types (e.g., ground, air, naval, logistics, intelligence) and must provide realtime free play of all types of orders. They are used to maneuver units, call for fire, provide tactical air support, perform engineering tasks, control convoys, and so on as the situation develops. Role players use these work stations to provide combat support and combat service support. They must have considerable familiarity with the military issues involved but the computer expertise required of them should be minimized.

Other software systems, whether ABCS or other C4ISR devices or affiliated simulation models, need to be able to determine the state of the simulation at any time and to deliver orders to the simulation. In CBS, this is accomplished by the Master Interface.

Technical control workstations manage the operation of the simulation. Tech control activities include taking of checkpoints, monitoring for memory leaks, detection of non-fatal loops, adjustment of parameters that control the trades between performance and functionality, dynamic data modification, and diagnosis and correction of all technical problems that arise.

Senior control workstations ensure that the training experience is achieving its goals. Senior control has access to "magic" commands that allow, for example, reconstitution of units.

Behavior of Subordinates

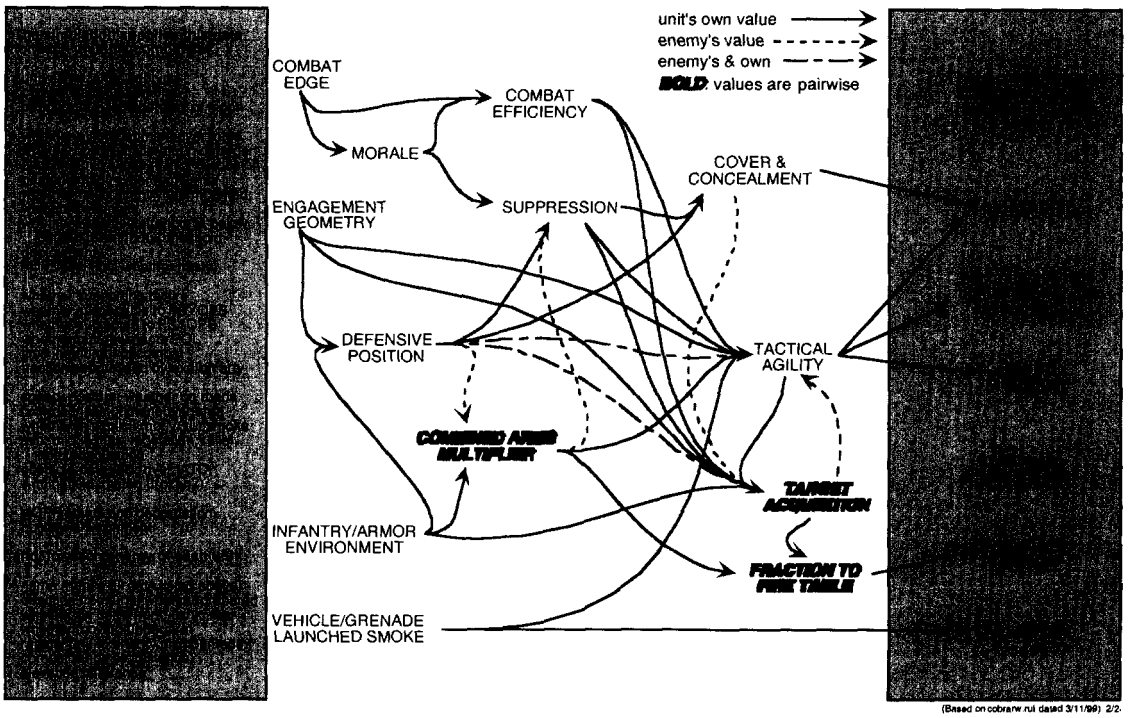
In the real world, subordinates should obey orders but be imaginative and responsive to unforeseen circumstances. In a simulation, whether it is used for analysis or training, subordinates should behave predictably.

In an analytical simulation with many repeated trials (so that statistical data is collected and used), the statistics of subordinates' behavior should reflect real-world expectations — if real-world expectations were known well enough to model accurately.

In a training simulation, in which there is one and only one run, subordinates must behave according to doctrine so that outcomes will be determined solely by the decisions made by the training audience.[8]

---

[8] This is not to say that there is no place for uncertainty in a training simulation. On the contrary, how one deals with uncertainty is one of the most important factors in combat.



CBS 1.8.0 COBRA Rule Set Dependencies

COBRA

Early in the development of the Corps Battle Simulation, General (Retired) Richard E. Cavazos, Senior Observer for the Battle Command Training Program, emphasized that training audiences must learn to deal with *everything* of importance on the battlefield, not just those things that are relatively easy to model, even if they are not modeled perfectly. Fatigue, thirst, morale, and suppression, for example, all affect how well soldiers fight. A commander who rotates his own troops, but makes it difficult for his enemy to do so should get better results than one who ignores the effect of prolonged combat. The issues that must be considered are legion. If the simulation ignores any of the factors, so will the trainee — at least during the exercise, as he is being trained to *win*.

Thus, the purpose of COBRA, a CBS acronym for *Combat Outcome Based on Rules for Attrition*, is to make it necessary for the training audience to take into account all METT-TC[9] factors in the tactical situation. How can a simulation account for factors that are beyond the scope of its existing models? In CBS, this was accomplished by implementing an expert system, with rules developed in discussions involving modelers from CBS, users from NSC, and a Senior Observer (General Cavazos) from BCTP. The approach was so successful that General (Retired) Edwin H. Burba, Jr., also a Senior Observer for BCTP and General Cavazos' successor on the COBRA Rules Committee,

[9] A mnemonic for the key elements to be considered in tactical decisions: Mission, Enemy, Terrain, Troops, Time, and Civilians. (See appropriate military training manuals for more detail.)

described COBRA as a "Theory of Combat". General Burba added the synergism of combined arms to the theory.

Not only is there nothing like this in the analytical world, most of these "soft" factors *should* not be considered in analytical models. Things that "cancel out" from a materiel acquisitions point of view can often be left out of an analytical model — and usually should be, as doing so simplifies the analysis without compromising its validity. If results in an analytical model depended on how long the troops had been in combat, the analyst would have to deal with unnecessary constraints (unless, of course, the analysis is designed to deal with that particular issue).

Analytical simulations require crisp models that produce precise quantitative results. Training simulations require comprehensive models that produce accurate qualitative results.

OPFOR

Since the OPFOR is not being trained, many functions, such as logistics resupply, can be implicit, resulting in a need for fewer OPFOR controllers. Yet, care must be taken. Tactically important OPFOR targets, such as supply lines, must be represented explicitly. This issue is unimportant in analytical simulations.

As with subordinates, OPFOR behavior in training simulations must produce doctrinal results so that outcomes will be determined solely by the decisions made by the training audience. For analysis, doctrinal behavior is sufficient. For training, however, OPFOR behavior must also pass the Turing test — that is, credibility demands that it must seem to the training audience that OPFOR is controlled by humans.[10] To put this another way: With the present state of the art of artificial intelligence, OPFOR must be controlled by humans to provide a sufficiently innovative and unpredictable opponent. Furthermore, credibility, which is needed for a valid training experience, requires that OPFOR must not be perceived as "looking under the covers" or *cheating* in any other way.

Gamesmanship

All players in a training simulation want to *win*. In some cases, careers are perceived to be on the line. Furthermore, many players see a training exercise as a game whose extended rules include the penalties for violating the nominal rules, rather than as an experience whose rules are the inviolable laws of nature. In training, *playing the game* should provide the desired training *experience*. There must be no benefit from actions that would be unrealistic.

There is nothing like this in the analytical world because analysts don't let weird things happen. No analyst uses a *withdraw* order to *attack* to avoid running into a strength threshold that would stop the attack. No analyst places an ambush behind a spurious ridgeline in the terrain model simply because it doesn't match the map. No

[10] If the state of the art of artificial intelligence reaches the point that it can be used to control real enemies, this will no longer be a requirement.

analyst would fill the battlefield with two-person observation posts because the simulation takes an entire attrition cycle to defeat *any* enemy unit. (CBS inserted an *overrun* model to deal with that one.) Training simulations must display *perceived truth* rather than *ground truth* on user workstations. A standard question at model design reviews for training simulations must be "Can this approach be *gamed*?"

Magic

On the other hand, the *exercise director* must be able to change essentially any conditions[11] in the simulation "by magic". It may be noted that providing this capability is sometimes tricky — consider, for example, the difficulty of magically moving a unit that is conducting a fire mission or engaged in direct fire combat.

As a bonus, magic capabilities make it much easier to set up test situations during development.

Artificial Intelligence

Suppose you *did* have an AI that was so good that you would put *it* in harm's way instead of risking soldiers. How would you use it in simulation?

Unless it routinely passed the Turing Test with an A+, so that its behavior was indistinguishable from human behavior, the first analytical studies you would perform would be to determine its doctrinal behavior as a subordinate and as an opponent.

In analytical simulations, you would use it over and over to get statistical results. Unless it routinely obtained those A+'s, however, you would not dare use only the AI in analyses, for results would fail to represent human threats.

You would *not* want to use it in training simulations even if you knew the enemy was going to field it. Rather, just as with human subordinates and human opponents, you would want to have the training audience experience *doctrinal* behavior — which would have to be expanded to include AI doctrine.

Would this conclusion still be true if the purpose of the exercise were to train the AI itself? No — provided you could play the stochastic game with the AI over and over and over and ..., so the AI could also learn from the *variability* of the experience, unlike a human student whose number of replications must be very small.

Dynamic Task Organization

Task organization is a very natural process for a training audience, but unnecessary in analyses.

---

[11] Magic capabilities must include resupply/reconstitution and movement, and should include everything else, such as: delivery of air-to-ground munitions, artillery, NBC contamination and decontamination, engineering tasks (laying and clearing minefields, building and destroying bridges, and so on), weather, PSYOPS, cooperation, morale, tactical agility, suppression, and even side affiliation.

Scaling

An analytical simulation can be tailored to the analysis being performed.

Training simulations, on the other hand, must be able to support exercises over a range of echelons. Typically, important phenomena occurs at significantly different echelons even within a single exercise.

ABCS / C4ISR Interfaces

This is another non-issue for analytical simulations — unless a particular ABCS[12] or C4ISR[13] system or an interaction among such systems is the object of the analysis.

In training, such systems are part of the interface between the training audience and the simulation.

Computer Performance

To produce statistically usable results, analytical simulations should be able to run at speeds greatly in excess of real time. Nevertheless, usefulness is not destroyed if high-speed performance is not achieved.

Training simulations *must* run in real time[14] during operations and significantly faster during recovery from a checkpoint. Some training simulations must be able to run in real time during periods of high activity (such as combat) and fast time during others (such as deployment).

Turnkey World-Wide Fielding

Analytical studies are usually conducted in a particular area with particular forces and equipment. The analyst can fine tune the simulation until the results are reasonable, then perform the analysis. For the next one, the analyst makes necessary changes to the model, retunes it, and performs the next analysis.

There is no such luxury for training simulations, as they are used by a large number of training audiences, often without resident support by the developers. They must run satisfactorily in playboxes located all over the world and, as noted above, produce credible results every time.

---

[12] Army Battle Command System.

[13] Command, Control, Communications, Computers, Intelligence, Surveillance, and Reconnaissance.

[14] That is, the ratio of delta elapsed simulation time to delta elapsed wall-clock time must be equal to 1.0 (or very nearly so). In fact, some ABCS and some C4ISR systems have no "understanding" of any clock rate other than real time. This fact can make the interface very "interesting".

Conclusion

Training simulations face many challenges that are either not relevant to analytical simulations or simply not very important in that context.

When used for analysis, simulations are used like analytical equations. Analysts control execution and run excursions until the study is complete. The emphasis is on getting comparative insights based on differences in input parameters.

When used for training, simulations are used as if they were a simplified (that is, doctrinal) version of the real world. Commanders on both side receive status and intel reports, make free-play decisions, issue orders, and respond to real-time outcomes. The emphases are on reasonable outcomes that reward good and punish bad command decisions, on realistic reporting, on real-time execution, and on simulation stability.

These differences could be summarized as: "Simulations *drive* analyses, *are driven by* exercises."

Thus, using the same simulation for analysis and training is not a realistic goal.

Where Do We Go From Here?

Neither continuing completely separate development nor forcing acceptance of algorithms "thrown over the fence" is the answer.

There needs to be universal recognition within the Military Modeling and Simulation Community that the Training part of that community has significantly different requirements than the Analysis part and that each has developed its own legacy of good simulations and useful models. We need to begin meaningful dialogue between training and analytical simulation users and developers. We need to strive for a variety of common algorithms for attrition, movement, detection, and so on.

California Institute of Technology

Analysis and Training — Why Different?

Joseph P. Provenzano
Project Manager, Corps Battle Simulation, JPL

Robert G. Chamberlain
Principal, CBS Modeler, JPL

Timothy Metivier
U.S. Army Combat Developer, NSC

71st Military Operations Research Society Symposium
Marine Corps Base, Quantico, VA
10–12 June 2003

Topics

- Historical Overview
- Credibility: *First* Time and *Every* Time
- Free, Dynamic Play
- COBRA
- Reliability and Recovery (Checkpoint, Modify, Restart)
- Dynamic User Interfaces
- Behavior of Subordinates
- OPFOR
- Gaming the Game
- Magic
- Other Differences
- Conclusions and "Where do we go from here?"

Historical Overview

Analysis

Experiments — Limited

Operations Research

Pencil & Paper

Mathematical Models

Map Wargames

Training

Field Exercises – Limited

Staff Rides

Command Post Exercises

Map Wargames

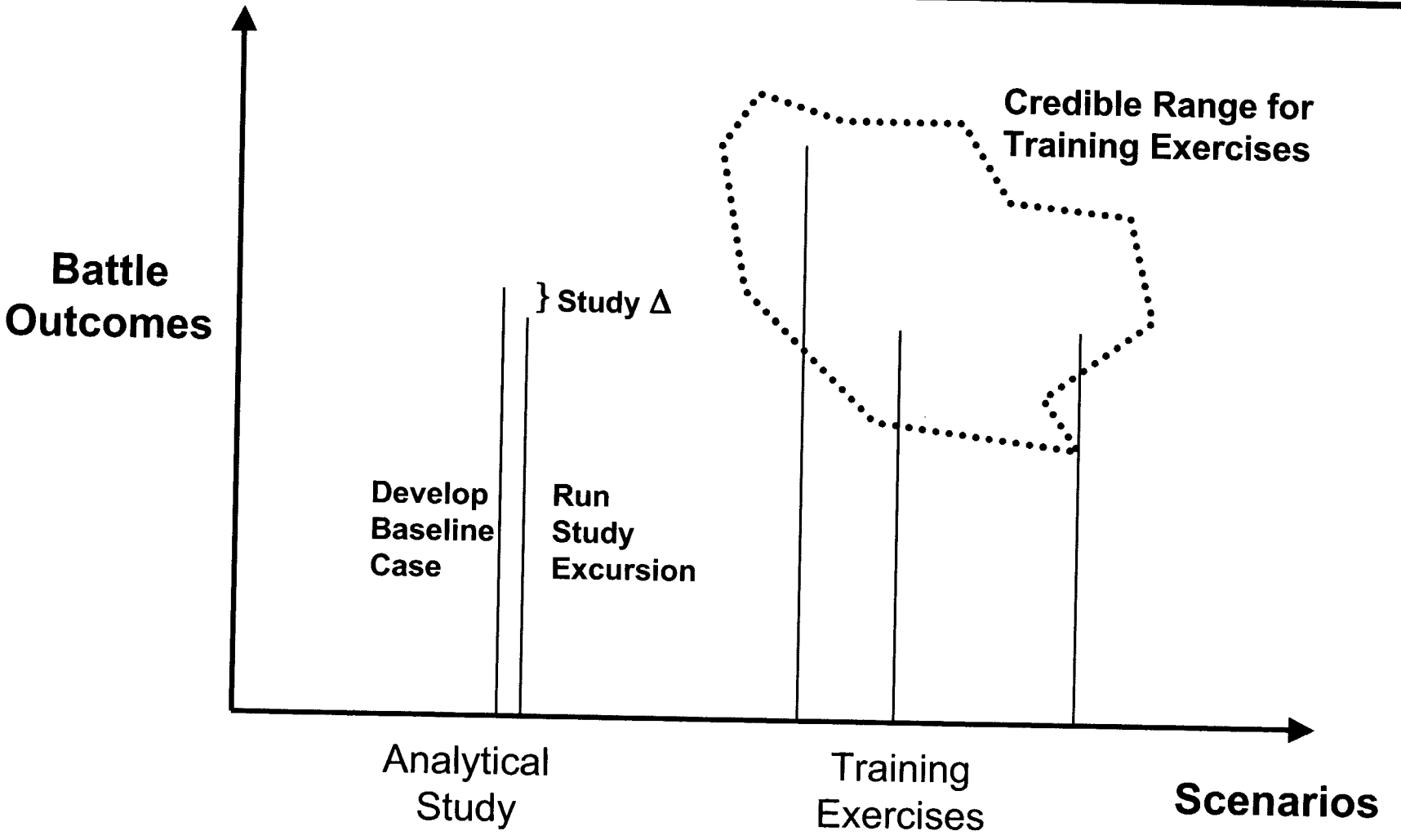
Sand Tables

Board Games

B.C. 1940 1950 1960 1970 1980 1990 2000 2010

California Institute of Technology

Credibility: *First* Time and *Every* Time



Free, Dynamic Play

Analyses: Controlled quantitative comparisons

- "Soft" factors undesirable
- Calibrated & tweaked during V&V
- Expect relatively small deltas
- Uncertainties produce statistical comparisons

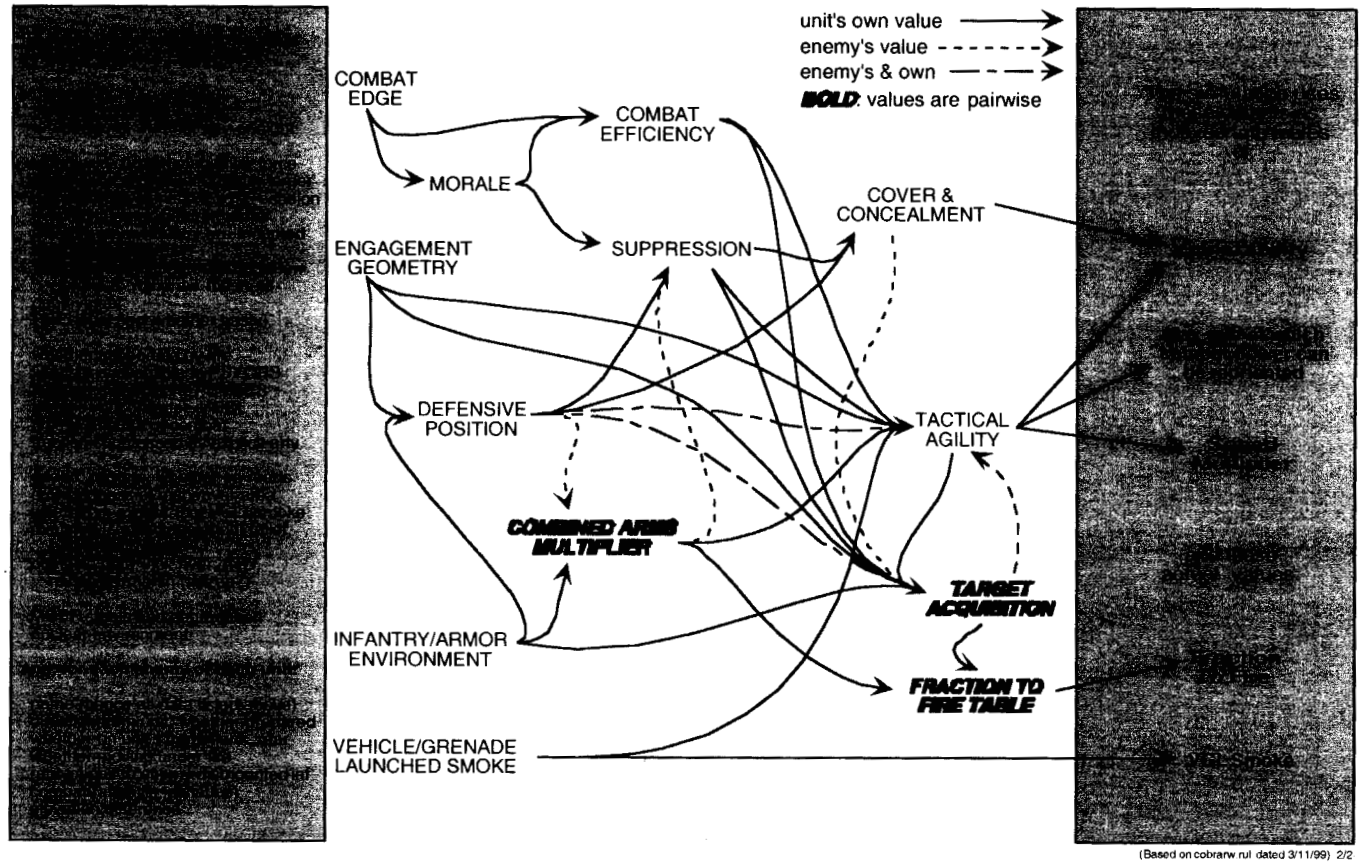
Training: Chaos

- Action—Reaction—Counteraction
  - » Requires free play with multiple players
- *Qualitative*, not *quantitative*, accuracy is extremely important
  - » Training stimuli must be good enough for decision making
  - » "Soft" factors crucial
  - » "Reward" good decisions with good results
  - » Allow — but "punish" — bad decisions
  - » Unimportant details must be credible

COBRA

Combat Outcome Based on Rules for Attrition

California Institute of Technology



(Based on cobranw.rul dated 3/11/99) 2/2

CBS 1.8.0 COBRA Rule Set Dependencies

Checkpoint / Modify / Restart
Reliability & Recovery

Recovery from regular checkpoints is essential during training exercises

- Training exercises run at 50+ sites non-stop for days
  - Even if the game stays up, may have "Game Over" too soon
    - » Due to a tactical blunder … or a tactical *coup de grâce!*
  - Even if no code errors, data may contain a "ticking time bomb"
    - » After several days of ops, *any* "computer error" could be a disaster
  - With a half million lines of evolving code, there *will be* errors
    - » Easy recovery is a better goal than impossible error-free code
    - » Also: Memory leaks, performance problems, functional bugs, or … functionality that is unacceptable to users in spite of how *good* it is
  - Either fix the code or disable whole areas of functionality
    - » Full or modular recompilation? How fast *is* that compiler?

Dynamic User Interfaces

Analytical simulations — typically, just the analyst/developer

Training simulations — many different types of users:

- Training audience (with ABCS / C4ISR) and OPFOR (and other sides)
  - » Role players (often know little about computers)
    - ❖ Ground, air, naval, logistics, intelligence, …
  - » Need perceived truth, not ground truth
  - » Free play
- Other software systems
  - » Linked simulations
- Tech control
  - » Take checkpoints, tweak performance parameters, fix problems
- Senior control
  - » Control the exercise, adjudicate disputes, perform “magic”

Behavior of Subordinates

In the real world: subordinates are taught to obey orders *but* to be imaginative in response to unforeseen circumstances

- If the commander is *lucky*, his subordinates will save his … command

In analysis: subordinates should reflect the real world to the extent possible for statistical analysis

In training: subordinates *should not* be imaginative

- *Negative Training would result if subordinate commanders were imaginative and made a bad plan successful — or vice versa**

* Really smart Artificial Intelligence (if it existed) would have to be used very carefully — if at all — in training simulations

OPFOR

OPFOR is a challenge for training simulation developers:

Level playing field is needed but there are many fewer OPFOR controllers controlling a comparable number of units

- Detailed, Symmetrical Modeling would require too many controllers
- Using modeling shortcuts for OPFOR can't preclude training audience options (e.g., resupply convoys must be on the road for targeting)
- Artificial Intelligence support is possible, but can't be perceived as an advantage for OPFOR (e.g., "smart chess player")

Gaming the Game

All players in a training simulation want to *win*

- Incentives are high — sometimes, careers are perceived to be on the line
- To some, an exercise is a *game* whose extended rules include the consequences of violating the nominal rules

In training, there must be no benefit from unrealistic actions
(In the analytical world, analysts don't let weird things happen)

We have seen:

Using a *Withdraw* order to attack—to avoid an automatic threshold

Using differences between maps and the terrain model for ambushes

Filling the battlefield with 2-man OPs because only one enemy could be defeated in a single attrition cycle (thence, CBS has *overrun*)

Magic

Although the *players* must obey the laws of nature, the *exercise director* must be able to "play God" to set the conditions for the commander's training objectives

- Magic resupply / reconstitute
- Magic move
- Magic engineer (mines, PDPs, bridges, on-order obstacles, ...)
- Magic NBC (contaminate and decontaminate)
- ...
- That is, magically change *any* conditions in the simulation

Magic is not always easy to code—moving a unit in combat could leave fire-support missions with nothing to support, detection processes falsely initialized, and so on...

Bonus: Magic makes testing much easier

Other Differences

Dynamic Task Organization

- Essential capability for training simulations

Scaling

- Multiple echelons are involved in training

Linkage to Other Models

- Real time, stable linkage is needed for training

ABCS / C4ISR Interfaces

- Real time linkage is essential for training

Computer Performance

- Real time performance is required for training

Turnkey Fielding

- Delivered world-wide for training

Summary

When used for analysis, simulations are used like analytical equations

- Analysts control execution and run excursions until the study is complete
- Emphasis is on getting comparative insights based on differences in input parameters

When used for training, simulations are used as if they were a simplified (doctrinal) version of the real world

- Commanders on both sides receive status and intel reports, make free-play decisions, issue orders and respond to real-time outcomes
- Emphases are on reasonable outcomes that reward good and bad command decisions, realistic reporting, real-time execution, and simulation stability

Conclusions

- Using the same simulation for analysis and training is not a realistic goal
- Neither continuing completely separate development nor forcing acceptance of algorithms "thrown over the fence" is the answer
- Need universal recognition that the training community has significantly different requirements and has developed its own legacy of good simulations and useful models

Where do we go from here?

- The M&S community needs a vehicle to begin meaningful dialogue between training and analytical simulation users and developers

- The M&S community needs to strive for a variety of common algorithms for attrition, movement, detection,...
  - "Soft" factors must be controllable
  - "Object-oriented" models?
  - Composable simulations?