Proceedings of the 1993 Winter Simulation Conference
G.W. Evans, M. Mollaghasemi, E.C. Russell, W.E. Biles (eds.)

THE CORPS BATTLE SIMULATION FOR MILITARY TRAINING

Sherry Mertens

Jet Propulsion Laboratory
California Institute of Technology
4800 Oak Grove Drive
Pasadena, California 91109, U.S.A.

ABSTRACT

The Corps Battle Simulation (CBS) is a discrete event simulation system that was developed for use in military training exercises. Development has continued for more than ten years. Early development emphasized increasing functionality, but recent versions equally emphasize increased performance and capacity along with interfaces to outside simulations. Early emphasis on software engineering aspects would have provided a more portable system today, but CBS still exemplifies how simulation technology can be used to enable more cost-effective training.

1 A CBS EXERCISE

The effort to cut costs throughout the military has increased the importance of simulated training systems. CBS provides the engine for training high level U. S. Army commanders and their staff. And, since CBS simulates the battlefield, the expensive use of combat troops, equipment, and land is not required.

During a typical CBS training exercise, a number of field command posts are set up along with a central simulation center. Each command post houses a unit's command and staff. The exact number of command posts depends on the level of exercise: it can be anywhere from brigade to corps, which equals 40 or more field command posts.

The atmosphere is kept as realistic as possible in the command post. The exercise runs in real time. The lighting, housing, and communications are as they would be in combat, and the staff performs their tasks just as they would in a real battle. They analyze the current battlefield situation, make decisions, then radio or telephone their commands to their subordinates located at the simulation center.

At the simulation center, there is a work station suite assigned to each command post. Each suite represents the training audience's unit, along with subordinate and supporting units. Additional work stations are used as technical control for the simulation. A suite typically includes two large color graphics monitors, a digitizing pad or mouse, a laser video disc player, a graphics display controller (GraphOver or Commodore Amiga 2000), a printer, and two or three computer terminals. Each computer terminal represents an individual work station and is operated by a controller. Each work station is assigned to a particular force (Bluefor, Opfor, or Neutral) and has a particular function (logistics, air, maneuver, senior, or technical). The controllers play the roles of the subordinates of the commanders who are being trained. In all, the simulation center involves approximately 200 people, including staff and support personnel, with the actual number depending upon the size of the exercise.

2 CBS DEVELOPMENT HISTORY

CBS is currently being developed by the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration. CBS is sponsored by the U. S. Army Simulation, Training, and Instrumentation Command and directed by the Army National Simulation Center.

CBS development began in 1983 under the sponsorship of the United States Readiness Command. CBS was originally named the Joint Exercise Simulation System (JESS). With the release of version 1.3 in 1990, JESS was renamed CBS. Throughout this paper, the name CBS is used regardless of which release is being discussed.

Development of a CBS prototype began in 1983. CBS was spawned from an early version of the Joint Theater Level Simulation. Version 1.0 of CBS was released in 1986. The first post-release use was exercise BOLD VENTURE 87 in November, 1986. CBS 1.0 was configured to simulate a three-division corps with an armored-cavalry regiment, a separate brigade, corps combat support and combat service support elements, and supporting air units.

Version 1.0 and other early versions focused on

enhancing and increasing functional capabilities. CBS 1.1, released in 1988, supported a larger playbox, added the workstation interface, and included new Army Aviation, Engineer, and Chemical models, and improvements to existing models. CBS 1.2, released in 1989, included improvements in the maneuver, air, and logistics functions, as well as the addition of standard routes in the workstation.

CBS 1.3 was released in 1990. It included functional enhancements in the Air Defense Artillery, Army Aviation, Airlift, and Artillery models. Also, the playbox was expanded to beyond three Universal Transverse Mercator zones. But version 1.3 was different in a very important way. For the first time, functionality was developed using a tool other than simulation. Combat Outcome Based on Rules for Attrition (COBRA) uses a rule-based expert system to analyze a combat situation. COBRA is discussed is more detail in Section 5.

CBS 1.3.5 was released in 1991. The development concentrated on more than expanding functionality. Although CBS 1.3.5 included several model enhancements, the primary focus was on increasing capacity so that the simulation could support up to 5000 units.

Increased capacity was again a focus with development of CBS 1.3R, released in 1992. This version was capable of supporting up to 7500 units. In addition, the development of 1.3R took on a third facet. This was the first version to provide a link between CBS and a second simulation system (Air Force Air Warfare Simulation) via the Aggregate Level Simulation Protocol (ALSP).

The current fielded version, CBS 1.4, was released earlier this year. It can support up to 20,000 units. Major functional enhancements were also made: detection of enemy units and perception of enemy units became more realistic and the capability of units to infiltrate into enemy territory was added. The upgrades in capacity and functionality for this version were a heavy enough burden on performance that it became a major focus. To help accomodate performance needs, major enhancements in the communications subsystem, executive interface, report generation, and the COBRA interface were made.

Additional functionality also places a greater burden on the controllers. To help alleviate the burden on the controllers, the Controller's Assistant (COAST) was developed as part of CBS 1.4. COAST is an automated controller for executing infiltration missions. COAST is further described in Section 5.

Development of CBS 1.5 has begun. A second link to the Combat Service Support Training Simulation System through ALSP is being added. The focus of the Armed Forces on simulation-to-simulation links means that performance will continue to be a major issue in future developments.

3 CBS DEVELOPMENT CYCLE

Ideally, a given development cycle begins before the previous one has ended. Representatives from the development team meet (and meet and meet) with representatives from the army. The meetings produce a set of enhancements to be developed in the upcoming version. The final set of enhancements is negotiated to reflect the best balance between the army's priorities and the size of the development staff (currently around 70).

Once a set of enhancements has been selected (and often even before they have been selected), a white paper is generated for each functional enhancement. The white papers are used as a basis for a workshop. At the workshop, developers meet with military representatives who are cognizant of the new functionality. Design issues are discussed as thoroughly as is possible this early in the development cycle. The workshop results in a complete set of requirements for each enhancement.

Once the detailed requirements are set, each enhancement is assigned to one or more modelers. The modeler takes the white paper and the requirements and turns them into a model design. The model design translates the military model into a CBS model. Communication with the military experts is essential to a successful model design. Better knowledge acquisition yields more realistic designs. And better model designs yield more complete software designs.

The model design is reviewed by a panel consisting of both developers and military representatives. The completed model design is then turned into a software design. Ideally, prototypes are being developed in conjunction with the software design documents so that all integration issues are worked out and incorporated into the software design. The software design is reviewed by development personnel.

After coding is complete, a code walkthrough is held. Then the developer completes testing the code before submitting it to the software baseline. A group of testers focus exclusively on testing functionality, writing trouble reports, and then testing the fixes.

Often, the model design is divided into a series of software designs. This helps to ensure a smoother transition into the baseline.

Before a new version is delivered and fielded, a functional verification is performed. The functional verification is run somewhat like a mini-exercise, though emphasis is placed on new functionality. The functional verification process has become much more complex with the addition of links to other simulation systems. Coordination with outside organizations is time-

consuming but very necessary throughout the development cycle.

Formal validation is performed by the military after a new version has been delivered.

4 CAPABILITIES

Executive control of the simulation is essential to a successful exercise. The technical controller can start, stop, and restart the simulation from a checkpoint. In addition to executive control of the system, certain "magic" orders are available to selected controllers through the workstation. Magic orders can instantaneously change a unit's location, supplies, and other state variables. Other magic orders can affect the simulation environment by modifying weather, terrain features, and engineering features.

The terrain data is based on 3 kilometer hexes. Hex interior characteristics include trafficability, vegetation, urbanization, roughess, and elevation. Hex edge characteristics include roads, rivers, bridges, and obstacles. Terrain data is available for a number of playboxes, including Korea, Europe, Central America, and Southwest Asia.

Units and logistical convoys move along the terrain. Movement may be either on roads or cross-country. Movement is affected by the moving entity's speed limit and march interval, equipment composition, posture, Mission Oriented Protective Posture status, and terrain. Moving entities may congest each other. The amount of congestion depends on the size of the moving entity, terrain throughput, and the presence of Military Police.

The bottom line results come from combat. In CBS units, convoys, and supplies can suffer losses through direct fire combat, artillery, and air-to-ground damage. Lanchester attrition principles are used to compute attrition due to direct fire combat. Artillery attrition can be caused by conventional, chemical, or nuclear munitions.

Fixed-wing air missions include Defensive Counterair, Close Air Support, Battlefield Air Interdiction, Air Interdiction, Offensive Counterair, Suppression of Enemy Air Defenses, Reconnaissance, and Air Refuel. A variety of missions may be combined into an air mission package. Rotary-wing air missions include Attack, Blocking, Reconnaissance, Support Screening, and Airlift missions. Both fixed-wing and rotary-wing aircraft may be used to airlift supplies. Air Defense Artillery weapons can detect and attrite aircraft.

Logistics elements of the game include non-battle supply consumption, maintenance, and medical models. Also, personnel are designated with a specific field expertise and assigned to crew systems.

The workstation provides a menu-driven, graphics-oriented user interface. Units, air missions, convoys, airbases, supply centers, obstacles, fortifications, contaminations, command and control lines, standard routes, and targets may be displayed at the workstation. Orders may be saved and reused. Reports may be saved and reviewed.

5 SOFTWARE ARCHITECTURE

The CBS software system architecture consists of seven major subsystems: The Game Events Executive Processor (GEEP), Workstation/graphics (WS), Master Interface (MI), Technical Control Station (TCS), database (DB), Controller's Assistant (COAST), and Combat Outcome Based on Rules for Attrition (COBRA).

The GEEP consists of an ever-growing group of 2800 SIMSCRIPT subroutines (approximately 250,000 lines of code), along with a few subroutines in C. The GEEP is the engine for the entire simulation system. The GEEP communicates with the other subsystems through the executive interface. Through the executive, orders enter the GEEP from various sources. The orders are processed, time is updated, and events are executed. The GEEP provides the bottom line numbers on attrition of units' systems and supplies. The GEEP communicates to the other subsystems through game messages and report data.

The GEEP contains the major portion of the modeling functions. When it became necessary to update the combat model, however, simulation was not the most suitable environment. So the COBRA subsystem was developed. COBRA is a rule-based model written in OPS5. A rule-based environment was deemed most suitable because of the many complex interrelated factors that affect ground combat. COBRA enabled the preexisting force-on-force attrition algorithm to be updated with integrated METT-T combat factors (mission, enemy, terrain, troops, and time available). COBRA not only provides greater realism in the ground combat model; but also provides results in a format that can be easily understood. This may be used in post-exercise analysis, a process that has become a big focus in military exercises that use CBS. COBRA sends results to the GEEP in the form of GEEP-readable orders.

The GEEP and COBRA together represent the entire combat model. The workstation subsystem provides the interface between the model and the controllers. The workstation provides a pictorial view of the current combat situation to the controller. The controller sends orders to the GEEP through the WS menu. And the GEEP sends information to the controllers through report data that is formatted into a complete report and

displayed at the WS.

Like the WS, the Technical Control Station also controls the simulation, but in a different way. The TCS communicates directly to the GEEP through the executive command interpreter. Unlike other subsystems that run in a VAX/VMS environment, the TCS runs in a UNIX environment.

A third subsystem that can control the simulation is the COAST subsystem. COAST is an automated controller. Its purpose is to reduce the ever-increasing load on the human controllers. Through the WS menu, a controller can order COAST to control the infiltration of a battalion-sized unit. COAST sends orders to the GEEP to split the battalion into smaller infiltrating unit elements. COAST then monitors the progress, controls movement, and handles contingencies for the infiltrating elements. At the appropriate time, COAST sends an order to the GEEP to merge the smaller elements into a single unit and passes control of the unit back to the human controller.

Before an exercise begins, the GEEP is initialized with data from the main database subsystem. The DB contains terrain, unit, and game data. Additional unit data can be read in at any time during the course of the exercise.

6 CONCLUSIONS

Given hardware constraints, the Army's focus on linking CBS with other simulation systems, and the continued need for more functionality and greater capacity, performance will continue to be a focal point of future development. When CBS 1.4 breached the performance saturation point, performance was given its due respect, and a developer was assigned to focus exclusively on the performance issue. If the original CBS development team could have known that CBS would still be alive and growing ten years later, software engineering issues certainly would have been addressed with greater care.

Perhaps if software engineering issues had been addressed appropriately early on, we would not be as securely tied to the specific machines and operating systems that we currently are. CBS has grown so big that an attempting to release these ties will be an exhaustive effort. However, it is certainly something we are working toward.

Finally, coordination with other development organizations is certain to be an issue in future work. The army is pushing links with other simulation systems. This means that whenever new functionality and data are incorporated into CBS, it must be done with the army's entire confederation of simulations in mind. We are no longer working in a vacuum.

ACKNOWLEDGMENTS

I thank all the people who have been working on CBS since its early years and still find the work enjoyable. The knowledge I gained from seeking out their unique historical perspectives was essential to the completion of this paper and only served to increase my respect for this animal that we have been nurturing over the past ten years. Thanks especially to Will Duquette, Jay Braun, Joe Fearey, and Ray Pritzgintas.

This work was performed at the Jet Propulsion Laboratory, California Institute of Technology. This work was sponsored by the U. S. Army Simulation, Training and Instrumentation Command, through an agreement with the National Aeronautics and Space Administration. Technical guidance was provided by the U. S. Army National Simulation Center.

REFERENCES

Jet Propulsion Laboratory. 1991. *CBS 1.3 Analyst's Guide Volume 2, Logistics.* JPL Internal Document D-7849. Pasadena, California.

Jet Propulsion Laboratory. 1991. *CBS 1.3.5 Analyst's Guide Volume 1, Ground.* JPL Internal Document D-7849, Rev. A. Pasadena, California.

Jet Propulsion Laboratory. 1992. *CBS 1.4 Analyst's Guide Volume 3, Air.* JPL Internal Document D-7849, Rev. A. Pasadena, California.

Jet Propulsion Laboratory. 1993. *CBS 1.4 COAST User's Guide: Infiltration.* JPL Internal Document D-10329. Pasadena, California.

Jet Propulsion Laboratory. 1993. *CBS 1.4 COBRA User's Guide.* JPL Intenal Document D-7856, Rev. B. Pasadena, California.

Jet Propulsion Laboratory. 1993. *CBS 1.4 Executive Overview.* JPL Internal Document D-7848, Rev. A. Pasadena, California.

AUTHOR BIOGRAPHY

SHERRY MERTENS is a member of the technical staff at the Jet Propulsion Laboratory in Pasadena, California. She received her B.S. in Mechanical Engineering (1984) and her M.S. in Engineering Management Science (1988) from Wichita State University in Wichita, Kansas.