Proceedings of the 1995 Winter Simulation Conference
ed. C. Alexopoulos, K. Kang, W. R. Lilegdon, and D. Goldsman

A BRIEF REVIEW OF THE COMMAND FORCES (CFOR) PROGRAM

Marnie R. Salisbury
David W. Seidel
Lashon B. Booker

The MITRE Corporation
7525 Colshire Drive
McLean, VA 22102

ABSTRACT

The command forces (CFOR) program is implementing a new aspect of warfare simulation: explicit modeling of command and control. The program presents several aspects: (1) a concept of operations where command and control nodes populate the battlespace in the same manner as weapons systems; (2) an architecture where software simulation of command and control interacts with the battlefield through a set of common services; (3) a software design for the services that forms an infrastructure that integrates with underlying Semi-Automated Forces simulation; (4) a mechanism that facilitates automated integration of real world C2 systems with simulations; and (5) an implementation plan that integrates the efforts of multiple developers to produce a functioning multi-service command forces simulation.

The CFOR program has passed through the concept and planning phases and is being implemented. Lessons learned from progress to date are presented along with a plan for further development and integration.

1 BACKGROUND

The Command Forces (CFOR) project is a part of the Synthetic Theater of War (STOW) program, an Advanced Concept Technical Demonstration (ACTD) that is jointly sponsored by the United States Atlantic Command (USACOM) and the Advanced Research Projects Agency (ARPA). The STOW program is scheduled to support a USACOM exercise in 1997 where objects from each US armed service will interact with each other and with credible opposing force objects in a virtual simulation environment.

The STOW ACTD requires the representation of larger-scale and more diversified military operations in virtual simulation. A key element in achieving this goal is the ability to represent both fighting forces and their commanders in software. CFOR extends the current DIS architecture to incorporate explicit, virtual representation of command nodes, C2 information exchange, and command decision making.

2 CFOR CONCEPT

Extension of DIS to incorporate command and control is based on four fundamental tenets.

(1) Command and control can be represented in terms of the interactions and behaviors of command entities.
(2) The C2 process is an information flow process among command entities. As a part of the CFOR concept, the Command and Control Simulation Interface Language (CCSIL) represents the information exchanges between commanders.
(3) C2 information flow must be restricted by a faithful representation of real world communications. Information flow must be routed through command nodes compatible with the real world and subjected to battlefield effects. As with real commanders, virtual command decision makers will have access to information about the world through their sensors, information reported by subordinates through CCSIL messages, and CCSIL intelligence messages from superiors.
(4) The C2 decision process is represented in the individual command entities—the originators and recipients of information exchanges.

3 CFOR ARCHITECTURE

Under the CFOR architecture, a command entity may be represented in one of three ways (See Figure 1):

- a complex software application (the original goal of the Command Forces program),
- a traditional computer generated forces application (e.g., an abstraction of the platoon leader is embedded in the ModSAF application),
- a human working at his/her real world command and control workstation.

ARPA's CFOR program is working to build and integrate several examples of all three representations of command entities to create a robust and intelligent synthetic force for the STOW ACTD.

The DIS protocols define a common interface for each entity that attempts to ensure interoperability and consistent physical interactions on the virtual battlefield. Analogous requirements exist for the C2 interactions among entities. Therefore, a Command and Control Simulation Interface Language (CCSIL) has been devised to fill that role in CFOR.

3.1 Technical Reference Model

CFOR includes a framework or technical reference model (TRM) for command entities to provide a well-defined, common interface for all command decision activities. This TRM (see Figure 2) promotes interoperability and coherent C2 activity by providing a shared infrastructure, a common set of information and computing services, accessible through a well-defined applications interface.

The TRM is composed of three layers: Application Layer, Information Services and Utilities Layer, and Baseline Infrastructure Layer. A layered approach provides three specific benefits: 1) it provides a means of centralizing control over the baseline of doctrinal knowledge needed by the command entity applications; 2) it reduces command entity developers' efforts by providing common reusable software; and 3) it shelters the command entity developers from technology and functional enhancements in the baseline applications (e.g., ModSAF) and allows them to focus on command decision behavior.

- The Command Entity Application layer is where the command decision-making processes reside. Command Entity Applications may be fully automated software or C2 workstations operated by human command entities. All details about the actual implementation of a software command entity are under the purview of the simulation developer organizations; they are free to implement their own approach to making command decisions. Likewise, the adaptation of C2 workstations to the CFOR architecture is dependent only on the interface specification to selected modules with the Information Services layer. Workstation developers are free to decide how to display, massage, or augment the simulation data available via the Information Services layer.
- The Information Services layer contains services and utilities that provide the information needed to support command decisions. These services impose few restrictions on how to model the decision process. They avoid making any inferences or judgments that are the proper purview of command entities.

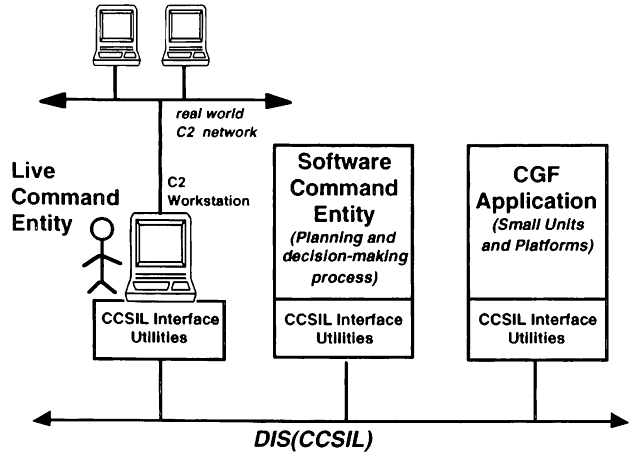

Figure 1: CFOR Architecture

Access to the services and utilities is implemented using an object-oriented, implementation-language-independent interface between command entity applications and the information services. To accomplish this, the Interface Definition Language (IDL) specification of the Common Object Request Broker Architecture (CORBA) was selected to define the interface and specify all interface parameters.

Services available include the following:

Platform Behaviors provide a generic interface to a command entity's physical representation on the battlefield. A command entity is associated with a vehicle

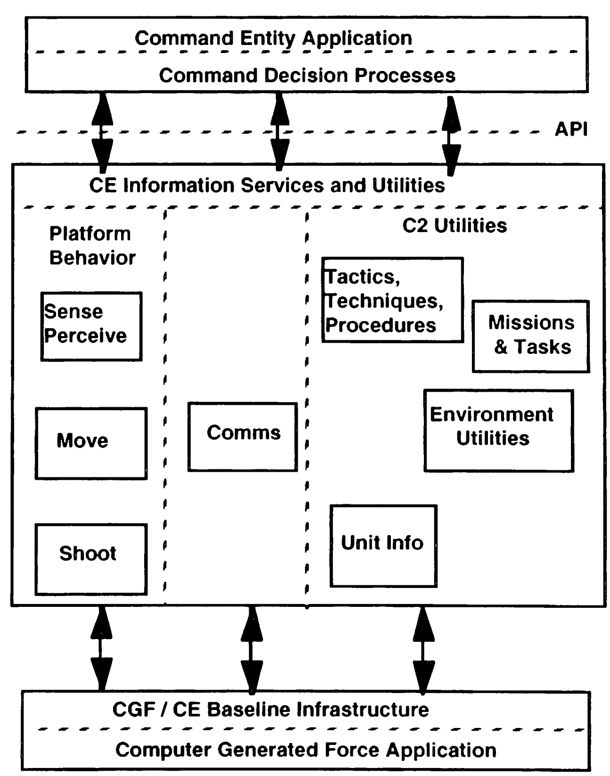

Figure 2: Technical Reference Model

or a set of vehicles (e.g., a command post). For example, an Army Company commander may ride in a tank, a Bradley Fighting Vehicle, a helicopter, or a HMMWV. Services provided mimic the commander's ability to sense from his vehicle, move his vehicle around the battlefield, and employ his weapons.

Communications offer an application interface to CCSIL message utilities.

C2 Utilities represent the background knowledge and rote reasoning capability of the commander—"routine" knowledge, shared by every human commander, that does not depend on subjective judgments. This is important for several reasons:

- To prevent redundant and potentially inconsistent knowledge acquisition and engineering efforts by the command entity developers.
- To help focus the activities of the command entity developers on addressing the difficult issues in modeling subjective, context-sensitive judgments and decisions.
- To localize the encoding of doctrinal information within the CFOR family of application software for two reasons: 1) to facilitate CFOR testing and evaluation; and 2) to minimize the effort needed for future enhancements or modifications for particular exercises or scenarios.

Services include

Environmental Utilities which provide the ability to compute mobility corridors, control measures, reverse slopes, routes, travel time and speed. (Environment includes terrain, ocean, and atmosphere.)

Unit Info which provides access to static data about units (own and enemy) and the ability to make basic inferences (e.g., combat power) from the raw data.

Missions and Tasks which provides doctrinal decision templates to help interpret an ordered mission and to devise a plan.

Tactics, Techniques, Procedures which provides templates to help fill out orders and implement a plan.

- The Baseline Infrastructure Layer contains the basic platform representation and general DIS interface utilities. These capabilities are accessed by command entity applications indirectly through the Information Services layer.

3.2 CCSIL

The Command and Control Simulation Interface Language (CCSIL) is a special language for communicating between and among command entities and small units of virtual platforms generated by computers for the DIS environment. CCSIL includes a set of messages and a vocabulary of military terms to fill out those messages. It was developed to facilitate interoperability between different implementations of command entities and platform entities (vehicles) in the DIS environment.

A common language designed for interpretation by software is needed to allow all three implementation approaches (workstation, automated command entity, and SAF) to work together in one environment. By using the structured format of CCSIL messages, humans at real world command and control workstations can send orders and directives to software command entities and expect them to react appropriately. Likewise, software command entities can exchange messages with each other.

Without a common language and communications services, every new element added to a DIS exercise would need to be iteratively retrofitted to interoperate with every other existing element of the virtual simulation environment. CCSIL serves as a unifying thread among diverse implementations of command entities, computer generated forces, and command and control workstations.

4 INTEGRATION WITH COMMAND AND CONTROL SYSTEMS

"Simulations should be driven by military personnel using their go-to-war C2 systems." This requirement is expressed routinely by the military user community. Until recently it has been difficult to meet this requirement. In limited cases, special automated links have been developed to connect a particular C2 system with a particular simulation. However, these point solutions are not generalizable to other C2 systems or other simulations.

The ability to interface C2 systems with simulations presumes several characteristics of computer simulations:

- Users at real world systems must be able to communicate with simulated counterparts. This means that simulations must represent information exchanges internally in a way that is functionally compatible with the real world and that simulations must include representations of command functions for the real world users to communicate with. Most combat simulations do not include representations of either C2 information exchange or command entities carrying out the command and control process to produce behavior in the unit. Often combat units are manipulated in the simulation environment as conglomerates using a set of abstract "orders" with no real world analog.
- Systems must resolve the inherent incompatibility between the way people exchange information and the way computer simulations can accept and interpret information. Humans use natural language which is rich, but fuzzy. Computer simulations require precise terminology organized in highly structured forms.

The CFOR concept and CCSIL bring a new approach to the construction of simulations that address this problem.

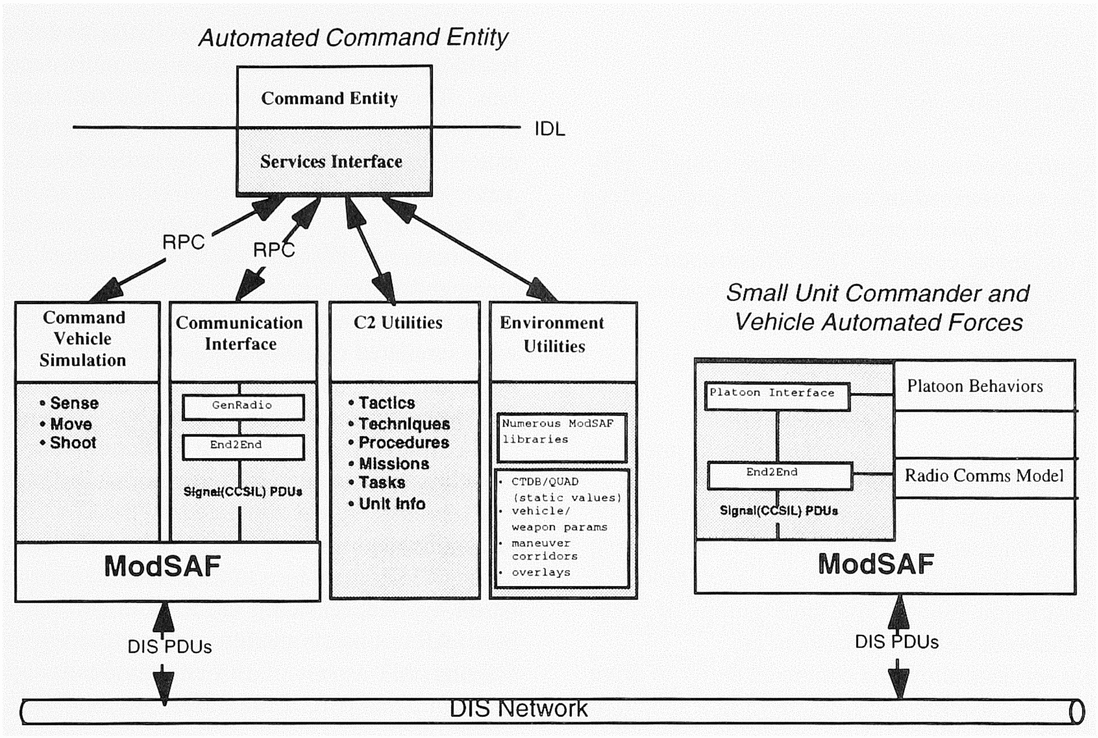

Figure 3: CFOR Infrastructure Software Organization

First, they provide an explicit representation of command entities and information exchanges; the CFOR concept provides a more appropriate simulated entity for a human operator to communicate with. A CFOR command entity collects and reasons on simulation information that is in a form appropriate for the human operator. The CFOR command entity can reply to requests for tactical state information, as well as, administrative and logistics information.

Second, CCSIL was designed to be both interpretable by software and to be a valid abstraction of the information exchanged by battlefield command entities. The current set of CCSIL messages focuses on providing structured, yet flexible formats for the types of information normally conveyed using natural language. The vocabulary of CCSIL messages was selected to coincide with the vocabulary of military personnel. The definitions and semantics for CCSIL vocabulary originally gleaned from field manuals and refined to reflect common military usage. Although it is not natural language English, it is much more robust than abstract simulation instructions like "Move-Unit" or "Attack".

One element of the CFOR program is to use CCSIL and the CFOR infrastructure services software to adapt existing real world command and control systems so that they can interoperate with the simulation components: the CFOR software command entities and the computer generated force representation.

5 CFOR SOFTWARE DESIGN

CFOR is supported by software. The software developed to date supports command entity application developers contracted by ARPA to build Army commanders. It will be augmented to support command entity development for the other military Services.

The software contains three components: adapted C2 workstation applications, infrastructure services software, and adapted computer generated force applications.

5.1 Adapted C2 Workstation

Work is underway to acquire and evaluate two Army C2 devices: the B2C2 and the Applique system. The selection of a C2 device for a Battalion workstation in the CFOR implementation for STOW-97 is based on availability (B2C2 exists as a prototype or beta-release only; Applique has not been released yet).

Once a system is selected from among the candidates, the workstation will be adapted for use in the CFOR simulation environment. The overall task is to link the workstation to the Information Services software to accomplish three things:

- Display incoming messages from simulated entities
- Provide a message creation tool so that users can create and send messages to simulated entities
- Update two-dimensional map displays and other databases used by the C2 workstation with data re-

ported by simulated entities.

5.2 Infrastructure Services Software

The CFOR infrastructure services software contains several modules as described in the CFOR technical reference model; they provide commonly used functions to the CFOR command entities. This infrastructure consists of layers of software organized into libraries, following the programming practices of ModSAF.

5.2.1 Platform Behaviors Module

Platform Behaviors Services provide a generic interface to a command entity's physical representation on the battlefield. The Platform Behaviors Services use the basic behaviors implemented in the computer generated force application in the Baseline Infrastructure Layer (ModSAF in the current application). The three groups of functions in the Platform Behaviors Services are

- *Movement* services allow the command entity to drive his vehicle to a specific location, drive in a specified direction, follow another entity, change the speed his vehicle is traveling, and change the orientation of his vehicle.
- *Sense* services allow the command entity to use the full range of sensors on his vehicle to sense other entities around him or to sense distinguishable terrain features around him.
- *Shoot* services allow a command entity to fire at a target, fire at a location, and fire in a sector.

5.2.2 Communications Module

The Communications Module helps a command entity application send and receive CCSIL messages. It offers an application interface to the following CCSIL message utilities:

- A message dispatcher that maintains a queue of incoming messages waiting to be processed.
- A notification mechanism that responds to polling by command entity for new messages.
- A message queue accessor that allows command entities to retrieve incoming messages from the queue.

A new feature that the Communications Module brings to the DIS environment is a capability to insure delivery of a DIS Signal PDU from the sending unit's computer to the receiving units' computers. The capability uses an acknowledge and retransmit scheme to insure delivery of the Signal PDUs containing CCSIL messages. As the DIS protocol evolves and multicasting services become available, this feature will be removed.

Note that this insure delivery feature is not the same as insuring delivery of the message between two command entities in the simulation. Realistic modeling of real world communications devices is a multi-faceted problem. The Communications Module software provides one piece of the large problem of simulating communications devices. It compares the radio identifier and frequency on incoming messages with the radio identifiers and frequencies to which they are tuned for the units being simulated. If any of the units simulated have radios tuned to that frequency, then the message is passed along to the unit. Otherwise the message is discarded. In this way, simulated units listening to the wrong communications net will miss messages broadcast on the net that they were supposed to be listening to.

The remaining aspects of communications effects modeling (propagation loss due to jamming, geography, and weather) are not simulated by the CFOR infrastructure software. Rather, the Communications Module hands CCSIL messages to the radio models in the simulation application being used. The current version of ModSAF has no simulation of communications devices. Solving this aspect of the communications modeling problem is not part of the CFOR program.

5.2.3 C2 Utilities Module

C2 Utilities provide command entities access to "routine" knowledge. This capability is implemented using a collection of software modules that have an input parameter list and return one or more data structures of information. The information is generated using basic data retrieval operations and simple assessment functions. These services have been designed to avoid making any inferences or judgments that should be made by the command entities themselves.

The C2 Services are organized into four subject areas: Environment Utilities; Unit Information; Tactics, Techniques, and Procedures; and Missions and Tasks.

5.2.3.1 Environmental Utilities

The Environmental Utilities (EU) provide an interface to S1000 terrain data that supports automated decision making. The utilities focus primarily on factors affecting movement of vehicles, cover, and delivery of fires for lower-echelon units.

The S1000 data format, used for SIMNET and ModSAF simulations, is efficient for real-time graphical display of terrain, but does not directly support automated command entity reasoning.

5.2.3.2 Unit Information

The Unit Information utilities assists in providing command entities with part of the minimal body of information expected of all commanders, no matter what their branch, experience, or expertise. These functions fall generally into two categories:

- Static information found in the battle books, field manuals, and technical manuals that are available to commanders in combat and training situations. This includes unit sizes and compositions, weapon and vehicle data, and estimated times to complete tasks.
- Unit assessment functions, that aggregate raw data into commonly used terms. For example, when a commander wishes to report the location of his unit, he can call a function with the location of each subunit in his command.

5.2.3.3 Missions and Tasks

The Missions and Tasks services provide a command entity with a skeletal decomposition of standard Company Team operations into tactically meaningful components, along with guidelines for implementing the tasks and subtasks associated with each component. The rationale for providing such a representation of doctrine is that a command entity competent in executing all the basic components can execute any mission defined in terms of those components. The components for Army Company teams build on the ARTEP collective tasks. The target repertoire of mission decompositions includes those missions corresponding to the set of virtual training exercises (Attack, Defend, Delay, Movement to Contact, Reconnaissance in Force, Raid, Exploitation, and Pursuit).

5.2.3.4 Tactics, Techniques, and Procedures

The Tactics, Techniques and Procedures (TTP) services provide a command entity with doctrinally acceptable decision options for conducting an operation. These services present tactical considerations and techniques, standard operating procedures, and "tricks of the trade" in a manner that facilitates the "how to" aspects of a commander's job. The decision options offered represent "textbook" solutions that every human commander would recognize from his military education. The motivating rationale for TTP services is to help command entities in those areas where human commanders can routinely generate an acceptable solution, regardless of their level of competence.

5.3 Adapted ModSAF

Adapted ModSAF is the current CFOR implementation of the Baseline Infrastructure Layer of the architecture. Five libraries have been added to make it CCSIL-compatible—to enhance ModSAF so that ModSAF units receive and react to CCSIL messages from command entities, as well as construct and send CCSIL messages to command entities.

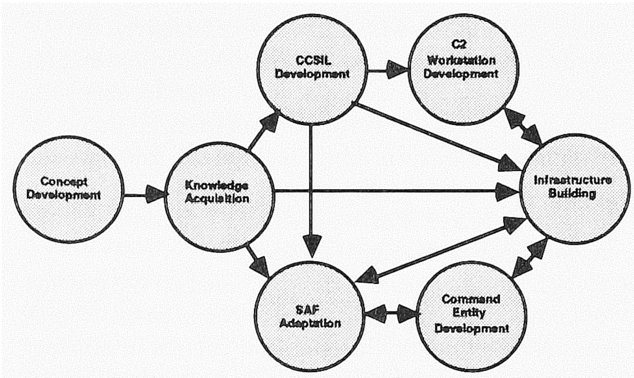

Figure 4: Activity Relationships

The Communications Module of the infrastructure software is used to send and receive all CCSIL message types. However, the adapted ModSAF can only respond to and react to the following subset of CCSIL messages:

- Operation Order
- Fragmentary Order
- Execute Directive (a subset of the complete set of messages)
- Report-Request

A CFOR Armor Company containing three tank platoons plus the Commander's tank and the Executive Officer's tank can carry out a subset of the doctrinally prescribed tasks for Tank Platoons (ARTEP 17-237-10-MTP) and Armored/Mechanized Companies (ARTEP 71-1-MTP). ModSAF's tank platoons and armored and mechanized companies can perform fundamental fire and movement tasks. More abstract tasks (e.g., "assist passage of lines") are not supported as well. Other tasks that are not supported involve interactions with the terrain that are either vaguely supported or not supported at all in the DIS environment. As solutions are found to these problems, ModSAF units will perform more doctrinally prescribed tasks and CFOR command entities will take advantage of more capable subordinate units.

6 CFOR DEVELOPMENT PROCESS

The process for developing the ultimate CFOR system is depicted in Figure 4 and described in the following paragraphs.

The process is being applied to each of the services independently, although oversight over the entire program is being applied by the program System Engineer.

- *Concept Development.* The first step in implementing CFOR is deciding and documenting which C2 elements will be represented in simulation and how each of them will be implemented (human, automated commander, or SAF). This concept is developed in close coordination with Service representatives.

- *Knowledge Acquisition.* Experts in each field and for each military Service gather information about the command process. Particular emphasis is placed on planning, decision-making, monitoring, and revising plans. After initial gathering and documenting by contractors, the Services will assume responsibility for maintenance of the knowledge base.
- *CCSIL Development.* CCSIL is based on the product of knowledge acquisition—on the documented C2 process and the identity, format, and content of relevant message exchanges. The CCSIL development team works closely with the knowledge acquisition team to assure clarity and completeness.
- *C2 Workstation Adaptation* Beginning with an Army C2 workstation (B2C2 or Applique), selected examples of C2 systems will be adapted to work in a virtual simulation exercise. The C2 systems will use the CFOR infrastructure services software to send and receive CCSIL messages and to control the physical portrayal of the commander in the virtual simulation environment (e.g., to move the command post from one location to another).
- *SAF Adaptation.* ModSAF is being enhanced to model new vehicles and small units and to model new behaviors for entities and small units. Currently this version of ModSAF is then adapted to properly carry out CCSIL orders and requests and to generate CCSIL reports. It is planned that the CCSIL adaptation will be integrated into normal ModSAF development.
- *Command Entity Development.* The CFOR program plan calls for multiple contractors, each developing a software implementation of a command entity. After a suitable period of development, the implementations will be evaluated. Subsequently, the developers will deliver new and improved command entities every six months until the 1997 demonstration. It is expected that initial experience will be gained in implementing Army command entities and that experience will be applied to implementing those of the other military services.
- *Infrastructure Building.* The CFOR infrastructure provide services to the command entity simulation and the real world C2 systems based on information provided by the knowledge acquisition process. An initial delivery of this software was made in January 1995; new versions are issued every three to six months, accommodating new CCSIL messages and modifications needed by Command Entity developers.
- *Testing and Integration.* The nature of the CFOR program dictates steps beyond the normal testing process. Technical integration testing is needed to assure that all components communicate correctly. Also Command entity behavior must evaluated against reasonable behavior standards, initially by the knowledge acquisition teams and ultimately by Service experts.

7 CFOR DEVELOPMENT STATUS

As described earlier, most of the CFOR development work accomplished to date has been in the Army domain space. Some work has been completed for the other military Services. Using the general outline described in Section 6, the following paragraphs briefly describe the status of that work.

7.1 Navy

Navy CFOR concept development started in February 1995. After defining a general concept, the effort turned to knowledge acquisition to support CCSIL development for the Navy. This work is ongoing; currently the Navy CCSIL message set contains about 35 messages that cover the anti-air warfare, strike warfare, and anti-surface warfare components of the Navy's mission space. Information Services software is in place for sending and receiving these messages. The Navy Synthetic Force entity development team (i.e., the team building the simulation of Navy ships and aircraft and their elemental behaviors) is adapting the simulation to send and receive CCSIL messages and linking the simulation software to the Information Services software.

Knowledge acquisition to support command entity development as well as development of command entities will begin in FY96. Throughout the process, Information Services software will be expanded to support the needs of the Navy command entities (e.g., new environmental utilities, new unit information data accessors). Identification and adaptation of the appropriate C2 workstation(s) is also slated for FY96.

7.2 Marine Corps

Marine Corps CFOR concept development started in February 1995 and continues. Software command entity development will probably begin in FY96 with the infantry platoon commander. CCSIL development, including knowledge acquisition started in July 1995. This work is ongoing. A large portion of the Marine Corps CCSIL messages are derived from the US Army CCSIL messages. The CCSIL development task involves comparing the Army CCSIL messages to the Marine Corps needs and changing or adding where necessary. Information Services software is in place for sending and receiving most of these messages.

Preliminary knowledge acquisition to support command entity development is underway now and will expand during FY96. This work, in turn, will feed the Information Services software development task.

The workstation task for Marine Corps CFOR is problematic due to the unavailability of automated C2 devices at the lower echelons of the Marine Corps. More research is required to determine whether a candidate workstation exists or a contingency implementation is necessary. This work will start in FY96.

7.3 Air Force

Air Force CFOR concept development started in December 1994. The Airborne Control Element (ACE) was selected as the first candidate for software command entity development. After defining a general concept, the work turned to knowledge acquisition to support CCSIL development and command entity development. This work is ongoing.

The first phase of Air Force CCSIL development was a review of the existing Navy CCSIL messages that address the air component. The Air Force CFOR knowledge acquisition team compared the messages to the Air Force's needs and recommended changes or additions where necessary. The Information Services software is in place for sending and receiving many of these messages. CCSIL development work will continue in FY96.

Development of software command entities will begin in FY96. The knowledge acquisition tasks will continue and will begin to feed the Information Services software development task (i.e., C2 utilities and environmental utilities to support the ACE command entity).

The workstation task for Air Force CFOR was initiated in July 1995. It leverages other work accomplished by the Air Force's Electronic Systems Command (ESC) to build a software interface between the Contingency TACS Advanced Planning System (CTAPS) family of software and the Air Warfare Simulation (AWSIM). The result of this effort is the CWIC workstation—a participant sitting at a CWIC workstation augments the rudimentary mission information provided in the Air Tasking Order and creates CCSIL messages that initiate behavior in the underlying Air Force Semi-Automated Force (AFSAF) simulation.

REFERENCES

Dahmann, J. S., M. R. Salisbury, L. B. Booker, and D. W. Seidel. 1994. Command forces: An extension of DIS virtual simulation. In *Proceedings of the Eleventh Workshop on Standards for the Interoperability of Defense Simulations*, 113-117. Orlando, Florida.

MITRE Corporation. 1995. *Command and control simulation interface language (CCSIL) message content definitions*, McLean Virginia.

MITRE Corporation. 1995. *Command and control simulation interface language (CCSIL) usage and guidance*, McLean Virginia.

MITRE Corporation. 1995. *Command forces (CFOR) environment utilities application programmer's interface (API)*, McLean Virginia.

MITRE Corporation. 1995. *Command forces (CFOR) infrastructure interface definition*, McLean Virginia.

Salisbury, M. R. 1995. Command and control simulation interface language (CCSIL): status update. In *Proceedings of the Twelfth Workshop on Standards for the Interoperability of Defense Simulations*, 639-649. Orlando, Florida.

Salisbury, M. R. , L. B. Booker, D. W. Seidel, and J. S. Dahmann. 1995. Implementation of command forces (CFOR) simulation. In *Proceedings of the Fifth Conference on Computer Generated Forces and Behavioral Representation*, 423-430. Orlando, Florida.

Seidel, D. W., M. R. Salisbury, L. B. Booker, and J. S. Dahmann. 1995. CFOR approach to simulation scalability. In *The Electronic Conference on Scalability in Training Simulation*. The Society for Computer Simulation, Institute for Operations Research and Management Science.

AUTHOR BIOGRAPHIES

MARNIE R. SALISBURY is a Member of the Technical Staff at the MITRE Corporation. She is the task leader of the CFOR CCSIL effort and primary author of CCSIL. Ms. Salisbury has ten years experience in military command and control and battle simulation.

DAVID W. SEIDEL is project leader for the Command Forces effort at the MITRE Corporation. For the past five years he has supported DoD agencies in integrating military simulations. He has over fifteen years of experience in development and application of military wargames.

LASHON B. BOOKER received his Ph.D. in Computer and Communication Sciences from the University of Michigan in 1982. Dr. Booker joined MITRE in August 1990 and is currently a Principal Scientist in the Artificial Intelligence Technical Center. From 1982 to 1990, Dr. Booker worked at the Naval Research Laboratory where he was head of the Intelligent Decision Aids Section in the Navy Center for Applied Research in Artificial Intelligence.