Issues in Developing the Potential of Distributed Warfare Simulation

Steven C. Bankes

RAND | NATIONAL DEFENSE RESEARCH INSTITUTE

The research described in this report was sponsored by the Defense Advanced Research Projects Agency. The research was conducted in RAND's National Defense Research Institute, a federally funded research and development center supported by the Office of the Secretary of Defense and the Joint Staff, Contract No. MDA903-90-C-0004.

ISBN: 0-8330-1249-5

Published 1992 by RAND
1700 Main Street, P.O. Box 2138, Santa Monica, CA 90407-2138

R-4131-DARPA

Issues in Developing the Potential of Distributed Warfare Simulation

Steven C. Bankes

Prepared for the
Defense Advanced Research Projects Agency

RAND

Approved for public release; distribution unlimited

PREFACE

This report presents the conclusions of the Distributed Warfare Simulation project conducted during 1987 and 1988 for the Information Science and Technology Office of the Defense Advanced Research Projects Agency (DARPA), in RAND's National Defense Research Institute (NDRI). The NDRI is a federally funded research and development center sponsored by the Office of the Secretary of Defense and the Joint Staff. The project examined the prospects for distributed warfare simulation and the technical challenges that must be met if these prospects are to be realized.

The project conducted a Workshop on Distributed Warfare Simulation, October 27–29, 1987, at RAND in Santa Monica. The workshop was jointly sponsored by the Defense Advanced Research Projects Agency and the Defense Communications Agency. The workshop was attended by 85 participants from the armed forces, the Department of Defense, academia, and industry. This report reflects conclusions drawn from that workshop as well as from other sources, and has been updated to reflect developments in distributed warfare simulation between 1988 and 1991.

SUMMARY

This report discusses the opportunities and technical options associated with the use of distributed computer simulation to support wargaming (including its use for training) and analytic studies. Primary interest here is on the geographical distribution of players and other personnel; however, support for distributing access necessarily involves other aspects of distribution such as distributed computation and distributed databases.

Potential long-term benefits that could accrue from a fully developed capability for distributed warfare simulation include:

- Player participation in games at wartime locations or without leaving players' operational stations,
- Greater access to gaming for various special training audiences such as senior commanders, reservists, and foreign nationals,
- Promotion of interagency coordination via joint development of and participation in distributed games,
- Test and evaluation of command, control, and communications systems as part of games played using operational equipment,
- Support for games covering wide geographic and functional areas,
- Easier utilization of human players as part of analytic exercises for studies of weapon system effectiveness or operational art,
- More frequent as well as more elaborate games,
- Improved utilization of various resources such as models, databases, and computers,
- Promotion of the exchange of models among developers and users, and
- Support for the process of model construction.

Whereas the list of potential benefits is striking, there are significant technical obstacles that must be confronted if distributed warfare simulation is to realize its potential. Developing a capability for distributed warfare simulation (DWS) providing all these potential benefits would require:

- Distributed communications facilities with adequate bandwidth,
- Software infrastructure to support distributed databases,

- Software infrastructure to support distributed processing,
- Software engineering techniques to ensure model correctness and maintainability in a distributed environment,
- Standard wargaming communication protocols, data dictionaries, and submodel interfaces to promote model interoperability, and
- Advances in combat modeling to provide improved model content, improved means for understanding model behavior, and better visualization of model outputs.

The challenges in this list range in difficulty. Some may be met by adapting off-the-shelf technology, whereas others cannot be resolved at present and require basic research. Existing DWS systems benefit from the geographic distribution of players in wargames but in comparison with long-term possibilities will:

- Suffer from problems with the quality of models,
- Be much less flexible in the types of games supported,
- Be brittle and difficult to maintain,
- Not support model interchange or the process of model construction, and
- Be more expensive on a per-application basis.

Thus, it is important to distinguish technical problems central to promoting current approaches to distributed gaming (which will be referred to as near-term issues) from deferred or neglected problems that will need to be confronted eventually if distributed warfare simulation is to achieve its greatest potential (long-term issues). Near-term problems are those associated with providing access to existing gaming facilities and are primarily driven by hardware needs and upgrades to current systems. Long-term problems are associated with creating a shared infrastructure to support a wide range of uses and are dominated by software engineering and simulation modeling issues.

Economic and bureaucratic factors tend to focus attention on near-term problems. *However, if capital investment in near-term projects is to contribute to the evolutionary creation of a general capability for distributed simulation, then near-term enhancements must be congruent with eventual system requirements.* Poor design features of existing systems must not be locked in because of sunk costs. Efforts to identify characteristics of ultimate target systems and research into fundamental problems should be pursued concurrently with near-term development.

Without adequate support software, systems that distribute computations and data will be much more difficult to build and maintain. In addition to improvements in software infrastructure, advanced DWS systems will require standards for distributed development and modeling must be improved. Problems associated with varying levels of aggregation are particularly critical in this regard.

Distributed games have important liabilities compared with centralized games, particularly in regard to the informal communication available in face-to-face encounters. Additionally, distributed warfare simulation has significant associated costs whose impact on a per-application basis will depend upon the frequency and diversity of distributed gaming. Frequent gaming allows infrastructure costs to be amortized across many games, which allows per game costs to become arbitrarily low. Distributed gaming is most advantageous when it can provide additional capability.

Long-term problems in distributed simulation overlap problems needing solution to improve the quality of warfare simulations in general. Thus, DWS is best viewed not merely as a singular technology, but rather as one component of a wider range of applications that could benefit from the solution of a core set of technological challenges. It is possible that DWS can serve as a driver for addressing problems that also merit attention in the larger context.

The overall approach to DWS development should be incremental. Activities that should be undertaken in the pursuit of creating a general capability for distributed warfare simulation fall into the following general classes:

- Developing specifications and high-level designs for desired future systems,
- Creating distributed wargaming systems based on current facilities to support near-term applications and assessing the strengths and weaknesses of the resulting systems,
- Initiating development of prototype distributed warfare simulation systems and testbeds to support medium-term applications,
- Supporting research into advanced features for incorporation into long-term systems, and
- Supporting various combat modeling community building activities.

ACKNOWLEDGMENTS

The author would like to thank Patrick Allen, Stephanie Cammarata, John Cushman, Paul Davis, Iris Kameny, Marlin Kroger, Lee Pleger, William Schwabe, Randall Steeb, and Robert Worley for their many helpful suggestions and thoughtful comments, and the participants in the Workshop for Distributed Warfare Simulation, whose enthusiasm and knowledge contributed greatly to this work.

CONTENTS

PREFACE iii

SUMMARY v

ACKNOWLEDGMENTS ix

Section

1. INTRODUCTION 1

2. THE PROMISE OF DISTRIBUTED SIMULATION 3
   Improving Resource Utilization 3
   Improving Interagency Coordination 4
   Facilitating New Applications 5

3. TECHNICAL CHALLENGES IN DEVELOPING DISTRIBUTED WARFARE SIMULATIONS 11
   Overview of the Technical Problems 11
   Telecommunications to Distribute Users and e. Interfaces 12
   Distributed Models and Databases 14
   Distributed Development 16
   Standards for Model Development 21
   Quality of Models 23

4. INCREMENTAL IMPLEMENTATION 24
   Specifications, High-Level Designs, and Measures of Effectiveness 25
   Applications 26
   Development of Engineering Prototypes 27
   Community Building 27

BIBLIOGRAPHY 29

1. INTRODUCTION

Dramatic improvements in the capabilities of telecommunications networks and computing machinery are combining to allow the easy exchange of voice, pictures, and data between locations anywhere in the world. This internetting of the globe is radically diminishing the importance of geography for a wide range of human activities (Builder and Bankes, 1990; Bankes et al., forthcoming). Distributed warfare simulation (DWS) combines telecommunications with combat simulation to provide a geographically distributed audience access to warfare simulation. By allowing geographically distributed users to simultaneously share a single simulated world, advanced forms of DWS could in effect create a virtual reality that could be used for a variety of ends. However, determining the appropriate role for this technology and designing the eventual system both require careful evaluation of the needs of the defense community and the characteristics of the technology. In this report, I describe both the potential benefits—for training, readiness, interoperability, planning, and decisionmaking—and the significant technical challenges of DWS.

Simulations of warfare have found a variety of uses:

- Decision support,
- Studies in military science,
- Operational and strategic planning,
- Command post exercises,
- Procedural training, and
- Weapons system evaluation.

For example, when a simulation is embedded in a mock-up of the controls for a complex system (such as an airplane cockpit), the result is a simulator that can be used for training with lower risk and reduced expense.

Early discussions of DWS emphasized its use for training and exercises (Lyons and Cushman, 1987). In the later half of the 1980s, numerous experiments in the use of DWS for training were conducted. Notable pioneering examples include the Warrior Preparation Center's use of distributed simulation facilities to support command post exercises for corps commanders (Rehmus, 1987) and distributed exercises run for the Battlefield Commander's Training Program and

Joint Warfare Center using the Corps Battle Simulation (CBS) model (Kahan et al., 1989).

The most technologically advanced large-scale experiment to date was the SIMNET project conducted by the Defense Advanced Research Projects Agency (DARPA) in partnership with the United States Army. Begun in 1983, the SIMNET project's goal was to develop a large-scale network of interacting combat simulators. The initial plan was to allow tank crews at a simulator training facility to conduct simulated tank-on-tank engagements; later, SIMNET was extended for training of selected aspects of combined arms [tanks and armored personnel carriers (APCs), fixed-wing and rotary-wing aircraft], gaming at a higher level using programs called "semi-automated forces" to dictate the actions of individual vehicles based on aggregate commands, and geographically distributed gaming over long-haul communications networks.

The SIMNET project has been assigned to the Army, but DARPA continues to pursue extensions as part of the Advanced Distributed Simulation Technology Program. Among DARPA's goals is a "seamless" universal system for training in all combined arms and joint warfighting skills, a command planning and doctrine development facility, and a testbed for the evaluation of advanced weapons systems before they are built.

It is not certain that the goal of a "seamless" simulation serving a wide range of users can be practically achieved. Nonetheless, the technical capability to allow distributed communities of users to interact through a simulated virtual world opens a plethora of potential uses, while posing some significant technical challenges. In the remainder of this report, the promises and challenges of DWS will be examined in greater detail. Section 2 discusses at length the reasons why distribution is desirable and the range of applications that advanced distributed simulation systems could uniquely serve. Section 3 examines the required technologies and indicates technical challenges that must be met before particular capabilities can be provided. Section 4 draws conclusions for both technical implementation and management of future system development.

2. THE PROMISE OF DISTRIBUTED SIMULATION

The potential benefits of DWS can be viewed from two perspectives: as a means for providing wargames and simulations to a wider audience, or as a novel communications medium that creates radical new options for exploiting warfare simulation.

The promises inherent in the first and more conservative view are not insignificant. Building simulations and using them to support wargaming and analysis are expensive and important enterprises. The requirement that all components of a wargame or analysis be collocated limits system availability, increases costs for travel and housing, lowers the quality of some applications, and makes other potential applications infeasible. DWS relaxes this requirement and hence can provide substantial improvements in costs and system utilization.

The most significant value for DWS will come not from cost savings but from benefits resulting from being able to use warfare simulation for new purposes. In facilitating the creation of a virtual reality, DWS could bring about a revolutionary change in the ways the military does its job, both in peacetime and in wartime.

IMPROVING RESOURCE UTILIZATION

The components that make up a simulation fall into three classes: human resources, physical resources, and computer software.

- **Human resources.** Many individuals with a variety of skills may be involved in the creation and use of warfare simulations. This is especially true for simulation to support wargaming, when the classes of "users" may include programmers, modelers, analysts, players, controllers,[1] and spectators. The requirement that all of these individuals be physically present at the gaming facility makes gaming expensive and limits both the frequency of games and the kinds of games that are feasible. Distributed wargaming can provide for much wider participation in wargames, thus increasing both the frequency of games that can be conducted and expanding the potential audience. Distributed simulation is thus a

[1]"Controller" is a general term used here for any person who interacts with the game but is not a member of the training audience. Thus, controllers may include both response cell personnel and white team players.

technology that can multiply the effectiveness of wargaming for training. The use of wargaming for analysis could also be enhanced by allowing multiple geographically distributed analysts to simultaneously view the same simulation.

- **Physical resources.** Computer installations with high-speed machines, high-capacity data storage facilities, or other unusual equipment are too expensive to be placed at every location where they might be of use. Simulation models often consume large amounts of computer time and memory and require very fast machines if they are to be executed in reasonable time. Distributed access to special computer facilities is desirable so that a wider range of users can benefit. For example, the Warrior Preparation Center and the Naval Wargaming Center are unique facilities whose utilization could be greatly increased through the use of distributed simulation.
- **Computer software.** Simulations are built from various software components, including models, databases, human interfaces, standard scenarios, outputs from previous model runs, and assorted support software. Software can be exchanged among facilities through use of transportable media such as floppy disks or magnetic tape. However, distributed access to software through telecommunications provides advantages such as instant widespread availability of up-to-date versions, ease of version coordination of software used at different locations, on-line support, and earlier availability of preliminary software for review by geographically distributed experts.

IMPROVING INTERAGENCY COORDINATION

Distributed simulation is a novel communications technology that could allow individuals in separate locations to interact via the virtual world being simulated. This novel medium could help coordinate individuals or communities where conventional means might not suffice. There are many groups whose efforts suffer from lack of mutual understanding, communications, and cooperation. Joint wargaming facilitated by distributed simulation can improve this situation in several ways:

- Players can observe each other play and develop improved insights about how other entities might conduct themselves in a war;
- Coordination problems can be uncovered through gaming;

- Communication outside the game can be facilitated by the gaming experience;[2]
- Insights into other approaches to warfighting may be gained by exposure to models embodying those as assumptions. This exposure could come from playing a game based on such a model or from the process of interfacing models created by the several communities.

All of these benefits can occur from centralized gaming. However, costs of transportation to a centralized facility or the length of time that senior players would be away from their commands may preclude or limit centralized gaming. Thus, distributed wargaming can contribute by facilitating interaction through gaming of organizations or communities that might otherwise seldom be able to game together. Distributed wargaming could improve communication and coordination:

- Between the services,
- Among the allies,
- Between active duty and reserve components, and
- Between educational, research, and operational communities.

FACILITATING NEW APPLICATIONS

Distributed warfare simulation could make new applications possible and permit new uses for existing facilities. In this subsection, the following ideas are discussed: continuous training through distributed gaming, gaming at operational stations, wide-area wargames, improving coordination and interoperability through DWS, DWS for combat modeling, facilitating wider access to computing facilities, DWS for advanced command, control, and communications (C3) and distributed decision support, and possible future innovations.

Continuous Training Through Distributed Gaming

In a distributed wargame, the players would not need to leave their stations and travel to a central facility but could play from their home installations. If such a capability existed, it would be practical to

---

[2]Such collegial interaction will be less pronounced than in centralized gaming where the players can meet face to face. However, telephone calls or video teleconferencing can provide a forum for discussion of issues that arise during a game.

conduct more frequent as well as more elaborate games.[3] This is of benefit not only because practice makes perfect, but also because when there are many exercises, some of them can explore unusual or particularly challenging scenarios, which is not feasible when the exercises are held only occasionally. In this regard, wargaming for training could begin to be used as flight simulators are: trainees could be stressed with unexpected breakdowns and emergencies in order to promote creative problem solving and adaptability. When games are infrequently held there is a strong tendency to run the standard scenarios, both because they are the best-estimate cases and to avoid morale problems due to "unfair" games.

With DWS, exercises could last weeks or months, allowing the inclusion of slow processes (such as strategic lift) that have no impact in most current short exercises. Such exercises would require that games be designed to be played at the same time normal duties are being carried out, because games with durations of weeks and months could never be held if the personnel involved were unavailable for their regular duties during this time. Similarly, the difficulty in freeing senior commanders from their normal duties represents an enormous barrier to the general use of wargaming for senior players. Distributed wargaming can allow greater participation of senior command decisionmakers. This is especially true since senior decisions during long games tend to be infrequent. Senior commanders could participate for a few hours per day and still attend to many of their normal duties.

For many training purposes, a one-to-one ratio of game time to real time may seldom be ideal. However, continuous training would allow this ratio to be tuned to the needs of the training audience, free of constraints of audience availability.

Gaming at Operational Stations

Distributed simulation makes it possible for players to play from their operational stations using the same facilities they would in an actual war. This can be important for procedural training and for the detection of problems that otherwise might not be noticed until the middle of a crisis or wartime (for example, see detecting C3 interoperability problems, below).

[3]The special benefits that accrue from face-to-face interaction will probably mean that centralized gaming will continue, even in an environment in which frequent distributed wargames were being conducted. Thus, distributed gaming should not be viewed primarily as a means for reducing costs but rather as a source of novel benefits.

Wide-Area Wargames

Distributed wargaming can allow games to be staged covering wide geographic areas that would otherwise be impossible because of the expense and disruption of transporting all of the players to a central location. For example, distributed gaming can facilitate the participation of the real commands in theaterwide or even global wargames.[4]

Improving Coordination and Interoperability Through DWS

Coordinating operations among the various branches of the armed services is a difficult problem but is critical to the success of many aspects of modern warfare. Cross-service wargaming can bring potential coordination problems to light and provide a framework for the exercise, development, and refinement of coordinated plans. Distributed wargaming can facilitate wargaming among the services by allowing the players to play from home stations.

Even greater coordination and interoperability problems arise in allied operations. Distributed wargaming allows games with allies in which players of other nationalities do not need to come to a U.S. facility to participate. Distributed wargaming overcomes various parochial barriers to gaming with allied powers by facilitating truly multinational games. For a game to occur, the participants must agree upon the game assumptions, which may not be easy with some of our allies. Although distributed gaming does not offer assistance in overcoming this barrier, it may serve to emphasize the question. There are those who will argue that forcing conversations on difficult topics may be one of distributed gaming's most important contributions.

Various problems in the command, control, and communications system could be detected and solved by means of games that players play from their operational locations. This would require play in which players use operational equipment, including communications devices. The system could be subjected to a more realistic load through an interactive wargame than is possible through scripted exercises or

[4]Games conducted over areas with large differences in time zones may require novel styles of gaming in which play is not simultaneous. Where direct interactions are not possible, loosely coupled subgames can provide joint context and reveal coordination problems in the use of shared resources such as strategic lift.

tests.[5] Communications interoperability problems, capacity shortfalls, and other C3 problems that might appear only in realistic situations could be exposed by games in which the players must communicate over the same equipment they would use in wartime[6] (National Security Industrial Association, 1987). Other advantages of using operational equipment include:

- Cost savings,
- Better training on actual equipment, and
- Testing of the capabilities of the operational systems.

Distributed wargaming could provide year-round training exercises for reservists that would enhance the readiness of reserve or national guard forces.

DWS for Combat Modeling

The ability to execute models from various locations can promote a more gregarious exchange of models, including both mature products and those still in development. Mature distributed warfare simulation systems could ease the availability of models and databases constructed at facilities around the country or the world. Although there are problems that need resolution prior to the realization of this possibility, easy access to distributed models and databases could have a profound impact on the way in which analyses are carried out. The extremely high costs and long time delays associated with building models or databases are a powerful disincentive to conducting aspects of an analysis requiring modeling. When confronted with the need for a new model, database, or analytic tool, the analyst is faced with the prospect of developing it from scratch or importing one from elsewhere. Both of these options require time and manpower. The availability of a large distributed library of models and databases could provide means for conducting analytic proce-

[5]Note that this would involve two networks: the operational communications network and the network associated with the game. Simulating the communications network would not have the same virtues.

[6]Although this application holds the potential for providing important diagnostics for the readiness of the C3 system, problems associated with effects on the operational system must be addressed. These problems include allowing modification to the operational system to connect it to the gaming system, tying up the operational equipment during the game, and the danger of providing foreign countries with a rich source of signals intelligence.

dures that otherwise would not be possible. A DWS system could provide access to:

- Models built and maintained by various institutions,
- Databases maintained at distributed locations,
- Display and analysis software developed at other institutions,
- Scenarios worked through in previous studies, and
- Standard cases.

Distributed experts could interact on advanced DWS systems and jointly work on a problem by simultaneously viewing the results of a simulation run, greatly increasing the options for collaborative work.

Wide Access to Computing Facilities

For projects that require large amounts of computer time, distributed simulation could provide speed-up through the exploitation of remote computing facilities with idle resources.

DWS for Advanced C3 and Distributed Decision Support

In distributed games, commanders must solve joint problems from separated locations, just as in wartime. The shared experience of projected battles under various assumptions and options could be a powerful means for geographically distributed commanders to do joint planning. (Note that this requires fast running, believable models that do not yet exist.) Distributed wargaming could be a useful platform for the testing and evaluation of advanced designs for C3 capabilities. The distributed gaming system could also be used to develop and evaluate novel techniques to support distributed decisionmaking such as video teleconferencing or interactive tactical maps (for example, see Adelman et al., 1986).

DWS for Military Science

If DWS systems are to fulfill the promises listed here, the quality of the underlying models must be better than any available today. Such models, if they can be provided, would improve the state of military science. (Indeed, military science must improve if they are to be built.) Whereas the large potential audience for DWS systems could provide the leverage to pay for the development of superior models, validation issues must be confronted if this technology is to deliver on its promises. The recent application of DWS (and SIMNET in particular) to the recreation of battles such as 73 Easting (Adams, 1991; Berry, 1991) is an important first step.

Future Innovations

The ability to conduct many more games, involving many more people, more often, will facilitate applications and studies that were not previously possible. This will provide a rich environment for creativity in devising distributed exercises and studies. It is likely that among the most important applications will be some we do not now anticipate.

3. TECHNICAL CHALLENGES IN DEVELOPING DISTRIBUTED WARFARE SIMULATIONS

OVERVIEW OF THE TECHNICAL PROBLEMS

Consider the challenge inherent in the construction of a comprehensive wargame: a virtual war of global scale, with every weapon system and unit on earth represented, with player roles for every member of the armed services, and with automated activity for every entity lacking a human player.[1] Such an application would require substantial computational and communications assets. Multiple geographically distributed computer centers along with adequate terminal and display facilities would be required to support players. Sufficient communications bandwidth would be needed to interconnect all of the players and computers, and both land line and satellite facilities would likely be involved.

A global wargaming capability would also require computer software resources, including simulation models, databases, and software to support communications, graphical displays, and other needs. This software would necessarily be developed by different groups in different locations, run on different geographically distributed computers, and yet made to interact with minimal additional effort. The last feature is necessary if diverse organizations such as the various services or the allies are to be easily drawn into joint gaming. A joint capability requires software to be drawn from multiple sources and integrated.

A global wargame also would require many human resources: players, controllers, and various support personnel. Although the problems of human resource management lie outside the scope of this report, problems associated with the coordination of these personnel is relevant. Initial experiences with distributed wargaming have generally used centralized controllers. However, as the scope of games increases, the need to coordinate a distributed control team as well as the support personnel for the various distributed facilities will need to be addressed. From a technical standpoint, these needs have implications for both software and telecommunications.

[1]Whether or not there is a need to conduct single games of global scope, a capability sufficient to do so would be required to support frequent games of diverse character.

SIMNET comes the closest of existing systems to having the capabilities required for such a global wargame, so it is useful to consider some of its technical characteristics. The SIMNET architecture is based upon the use of parallel and distributed multiprocessing across a network of high-speed multiprocessors in workstations, dataservers, and embedded in simulators. Each processor maintains its own image of the world and broadcasts update packets across the network as appropriate. Consequently, the global simulation is the collection of models and databases on the various processors connected to the system. Both local area and long-haul networks are required to connect the various processors, and the software for the system is developed by different groups at different facilities. If we imagine extending the SIMNET system to global scale, the requirements for telecommunications, parallel and distributed multiprocessing, and distributed development will become correspondingly complex.

In the remainder of this section, various technical challenges are considered in greater detail:

- Telecommunications to distribute users and user interfaces,
- Distributed models and databases,
- Distributed development,
- Technical standards for DWS, and
- Improving the quality of the model.

TELECOMMUNICATIONS TO DISTRIBUTE USERS AND USER INTERFACES

Adequate telecommunications facilities are a constraint on the pace of implementation of DWS systems. However, the enormous expansion of available communications bandwidth that is under way (due in great part to the deployment of fiber-optic cable) implies that telecommunications will not be a significant barrier to DWS in the longer term.

For unclassified gaming, T1 grade lines are now or will soon be available at minor cost throughout the industrialized world. Connectivity to remote locations may require satellite links, which will be more costly. Classified gaming may be somewhat more constrained because of the need for secure communications. Realistic simulations of weapons systems can convey classified information. Furthermore, if a potential enemy were to examine the communications of players over numerous distributed games, he could learn sensitive facts about the way we would conduct ourselves in a war. Consequently, communications will need to be encrypted, and it may be necessary to take steps to prevent the examination of external features of message traffic as well.[2] This subject merits study as distributed wargames become more commonplace.

It may be desirable to conduct DWS communications over existing or planned operational data links. Such an approach poses technical difficulties, but could provide cost savings, secure communications, and advantages connected with the use of operational equipment in exercises.

The SIMNET system could in principle be run on any network that offers the necessary features (in particular, connectionless data transfer) and is sufficiently fast. At present, SIMNET runs across ethernet local area networks.

Requirements for network response time vary depending on what is being simulated. In a command-level exercise, delays of many minutes could be tolerated. However, aircraft pilots trying to fly simulators in close formation would require response times of less than a few tens of milliseconds. Response time requirements are not constraining for local area networks, but may impose limits on geographically distributed architectures relying on long-haul telecommunications. Transmitting by satellite channels, for example, imposes a delay of a few hundred milliseconds. The amount of delay acceptable to players varies greatly with the nature of the game. Such a delay might be acceptable for SIMNET tactical simulations involving only ground vehicles (Pope, 1989), but would be less compatible for simulations involving fixed-wing aircraft.

Computer networking for distributed simulation requires not only the physical interconnection of the various sites but also standard software to mediate the details of message handling. Computer networks are designed as a series of protocol layers, with each layer responsible for some aspect of the network's operation (Tanenbaum, 1981b; Pope and Miller, 1987). A basis for the description of network protocols is the Reference Model of Open Systems Interconnection (OSI) of the International Standards Organization (ISO). This reference model has seven protocol layers consisting of the physical link layer, data link layer, network layer, transport layer, session layer, presentation

[2]For example, spurious random messages could be transmitted during and between games to obscure the profile of message traffic generated by the game activities. This approach would increase the bandwidth requirement for the games, in that the maximum needed bandwidth would be used constantly.

layer, and applications layer. The ISO/OSI standard can serve as the basis for defining a standard wargaming protocol for distributed wargaming. Whereas the lower protocol layers can be adapted from other applications, the upper layers, particularly the applications layer (layer 7)[3] will require careful definition if flexible interconnection of a wide range of facilities is to be supported (see discussion of standards below).

DISTRIBUTED MODELS AND DATABASES

In principle, sufficient telecommunications capability alone would suffice to provide distributed access for geographically distributed users; however, more advanced DWS requirements will require distributing the computation and databases as well. Local computers are necessary for advanced interactive graphical interfaces at the various player sites. Local computation lowers communication costs and improves response times by taking advantage of computer power across the net.[4] Locating models and databases at the sites of the appropriate support personnel can ease problems of model and database maintenance.[5] Relaxing the requirement that a designated computer center must be on line for the game to proceed can improve system flexibility and robustness (although major exercises may still require dedicated central nodes supporting scarce expertise). Sufficient communications bandwidth also provides flexibility to allow for the adaptation and tuning of the system architecture by system designers and managers.

Compared with centralized systems, distributed simulations confront increased system complexity. Support software must keep track of the locations of all entities, make sure all data arrive at the correct destinations, and ensure that the state of the computation across the network is consistent. *Without adequate support software, systems that distribute computations and data can be much more difficult to build and maintain.* Potential benefits of distributing computation include communications bandwidth reduction and increased system flexibility.

---

[3]The SIMNET protocol (Pope, 1989) is an example of an applications layer protocol.

[4]Realizing this improvement in speed depends critically upon details of model design and the nature of the support software. It should not be regarded as an automatic virtue of distributed computation.

[5]It should be noted, however, that scattering support personnel that otherwise would be centralized has its own inherent problems.

The implementation of simulations that can be executed on distributed computers can be eased by the use of special languages supporting distribution[6] or by distributed operating systems.[7] Commercial software to support distributed computation is now becoming available.

Similarly, databases can be maintained at distributed locations (Ceri and Pelagatti, 1984). Today, *homogeneous* distributed database management systems are commercial realities.[8] Prototype implementations of distributed *heterogeneous* databases have been limited to read-only systems. Research addressing distributed update mechanisms are in progress (Breibart et al., 1987; Pu, 1987). Thus, application of distributed databases to simulation is not off-the-shelf and will require future development. Distributed database systems such as IBM's R* (Lindsay, 1987) and distributed Ingres (Stonebreaker, 1979) are among the first generation of homogeneous distributed databases. Because of their homogeneity, they have not had to address many of the challenges of distributed heterogeneous databases. In heterogeneous federated database systems, a collection of individual autonomous database management systems (DBMS) agree to share information and yet retain some amount of local control. This configuration requires cooperation among data management systems with differing data models, data definition and manipulation languages, transaction management, and internal data structures. An important research area in distributed DBMS is concurrency control of updates, which requires some means of synchronization among distributed nodes. SIMNET handles this by having each node be an object responsible for broadcasting its state change. Any node missing a broadcast maintains approximate consistency by dead reckoning from previous data. The importance of continued distributed DBMS research is evidenced not only by the recent growth in the field, but also by the announcement by some DBMS vendors of distributed

[6]These techniques allow the modeler to construct the simulation model without the details of how it is to be distributed (Bastani et al., 1987; Beckman et al., 1988; Fox et al., 1988; Jefferson and Sowizral, 1982; Marti, 1988; Scott, 1986b; Unger and Jefferson, 1988; Wah and Juang, 1985).

[7]Distributed operating systems (Berets et al., 1985; LOCUS, 1984; Popek and Walker, 1987; Scott, 1986a; Spector, 1981; Ward, 1980; Wulf et al., 1974) allow programs on different machines to interact as though they were resident on the same machine.

[8]Homogeneous distributed database management systems are implemented using identical hardware and software at all the nodes (distributed locations). The desire for flexibility together with political and bureaucratic realities can imply the need for heterogeneous distributed database management (i.e., computers from different manufacturers).

database products that include "gateways" to heterogeneous systems (Sarin, 1987).

DISTRIBUTED DEVELOPMENT

If comprehensive global-scale DWS systems are to be feasible, it will be desirable for "pieces" of the systems (i.e., models of particular subsystems, scenario definitions, simulators, and visualization tools) to be separately developed by different groups, at different locations, and at various points in time. Consequently, the greatest contributions of DWS can be realized only if there is support for distributed development and easy interoperability of simulation components. Work is under way toward providing a general basis for distributed development (Berets et al., 1985; Shatz and Wang, 1987), but a general capability for distributed development of DWS systems will require innovations in both the software engineering of the computer programs and the science of military modeling.

Distributed development will require meeting several significant challenges, but it will also provide important benefits. The process of constructing models is often driven by the interests of particular institutions, and models reflect the predispositions and attitudes of their developers. The use and promotion of particular models often become rather partisan. The ability to construct composite systems from preexisting subcomponents could provide options for dealing with such parochialism and promoting widespread participation in distributed wargaming. Even when parochialism is not an issue, there is a frequent desire to combine submodels with particular characteristics for specific games or analyses. Additionally, the ability to easily adapt pieces of models could result in substantial reductions in costs and improvements in quality through the reuse of model components.[9]

Distributed development would be a noteworthy advance even if the simulation itself were not distributed. However, because distributed development and distributed wargaming have similar overall goals, distributed development can quite properly be thought of as a long-term continuation in the development of DWS capabilities.

Unfortunately, a general capability for distributed development is technically difficult, especially when it involves the retrofitting of

[9]Such a capability does not now exist. Past attempts at such integration have failed because of problems of focus, inadequate software tools, and the lack of a variable resolution design.

models that were developed for different purposes. On occasion, separately developed models have been successfully combined, but only by extensive reprogramming and other labor-intensive practices. Such "siege" tactics will not be cost effective for general purposes and cannot succeed for systems above a certain level of complexity. Easy integration of submodels that were separately developed requires the *a priori* existence of technical support for the integration process. This is especially true when the integration must combine models that operate at various levels of aggregation.

In SIMNET, distributed development is promoted through the specification of a network protocol (Pope, 1989), a specification for database interchange (Wever et al., 1989), and by reliance on the grounding of all simulated objects in the "physics" of individual vehicles and their activities (such as exchanging fire). Such a shared conceptual model provides an assist in coordinating the efforts of distributed development activities. Shared frameworks are less available for more aggregated modeling (such as constructing semi-automated forces).

The barriers to distributed model development are closely related to a number of other problems in the science of simulation modeling. These barriers arise because there are unstated assumptions that lie behind the surface behavior of the model. Because the semantics of models are not explicitly represented, integrating two preexisting models can be extremely difficult. Even after syntactic conflicts have been resolved, there may remain difficult-to-detect conflicts between the meanings of the submodels.

The process of model integration can be divided into three stages:[10]

1. Making the models work together. This is the stage of resolving syntactic conflicts between the models.
2. Making the models work together correctly. This is the stage of resolving semantic contradictions between the models.
3. Making the models work together understandably. This involves ensuring that the semantic relationship between entities in different models can be traced without prohibitive complexity.

[10]In actual practice, these are not necessarily sequential stages but rather different aspects of the problem of model integration. The ordering here represents increasing levels of difficulty and will often be reflected in the order in which these problems are resolved.

The importance of stage three is often overlooked. The usefulness of a simulation model is greatly diminished if the causal relationships in a model run cannot be understood. Analytic use of simulation obviously requires understanding model outputs. Such understanding is also very important for training. Without it, one of the great advantages of nonscripted games is lost. The biggest opportunity for learning occurs when a player is surprised. Unless unexpected outcomes can be understood, the players may either ignore or reject those outcomes as unrealistic, or worse yet may take away false lessons learned because unrealistic results are not recognized. Although initial encounters with computer-driven wargames are often exciting for participants, we must anticipate that as such games become commonplace, the expectations of quality and understandability for those simulations will rapidly increase. The systems we build must be designed to meet those expectations if the full potential of the technology is to be realized.

To make the three stages of model integration more explicit, consider an example combining an air model and a ground model from different sources to create an air-land battle simulation.

In the first stage, the various interfaces between the models must be made to correspond. This may involve changes as trivial as spelling differences (CAS sorties vs. air strikes) or as difficult as resolving differences in resolution (one model might be hex based while the other uses avenues of approach). This stage is relatively easy in that by definition any model differences resolved here were explicitly represented. Until problems of this type are resolved, the models will not run together at all, or will produce some outputs that are clearly nonsensical.[11]

Problems of the second kind are more difficult to detect in that the combined models may produce results that look generally plausible but are nonetheless wrong. Again, there is a range in the difficulty of these problems. An example of a relatively easy semantic mismatch would be a difference in units of measurement. One model might produce a number in units of miles whereas the other expects it to be in kilometers. Or more subtly, the air model might consider weapons load to quantify all weapons, whereas the ground model uses it to mean air-to-ground weapons only. More difficult problems occur when the semantic conflict reflects differing assumptions, so that the differences do not attach to any particular item in either model. For

[11]The results may not be nonsensical for all cases, however, which can create difficulty.

example, the air model may have been developed for analyses of the air war and its impact on the ground war, tacitly assuming away any scenario in which the progress of the ground campaign affects that in the air. With no explicit interface or submodel to trigger examination, the lack of a routine to deal with this on the air model side might not be noticed until a game in which an air base, overrun by enemy forces, continues to generate missions. The most difficult problems of this class reflect differences in philosophy behind the modeling efforts, and may be hard to detect without extensive understanding of all the models involved. For example, the air model may allocate close air support under an assumption that NATO will defend forward until enemy breakthrough is imminent. If the ground model contains parameters presuming more flexible tactics, the model outputs may be plausible for all scenarios, but fundamentally invalid.

Problems of the third class are often ignored, but if unresolved can seriously reduce the usefulness of the resulting model. Model outputs, even when correct, are much less useful if the user is unable to investigate why they occurred.[12] For example, consider a run of an air model that shows Blue exhausting high-technology munitions on D+20 and Red achieving a breakthrough on D+24. To what extent did the first event cause the second? This question may be partially addressed by making additional model runs, but real understanding requires examining the histories of various model attributes, which may mediate effects between the two. If the means by which the user can make such an examination and the nature of the representations internal to the two submodels are such that understanding is impeded, then the usefulness of the joint model is greatly diminished. This is true even if the integrated model is a valid representation of the real world.

The foregoing gives an indication of the depth of the problems that need to be addressed to support the integration of separately developed models. These problems are sufficiently difficult that the retrofitting of existing models that were not designed to be interfaced can be undertaken only on a case-by-case basis, with large investments of manpower and potential shortfalls in model quality. Similarly, no technology exists that can guarantee the interoperability of arbitrary models.

---

[12]Of course, it's all in the computer so in principle any such question could be answered. However, if the level of effort required to do so limits the number of such questions that may be feasibly addressed, then the users will not be motivated to ask them.

However, distributed software development routinely takes place in which there is a preexisting global architecture and well-defined interfaces between the pieces that are to be developed separately. A trivial example is the utility of software tools such as compilers, standard graphics packages, and so forth. Such tools succeed because in addition to the software itself, there is a strong model of relationship among software components and an explicit specification for the interfaces between the parts, in the form of language reference manuals or other documentation. The advantages of this sort of decomposition have not accrued for arbitrary simulation submodels because of the enormous variety of possible models and interfaces. However, for restricted domains with regular structure and large amounts of modeling activity, such compartmentalization may be possible, and may provide real benefits in supporting software reuse and distributed development, as well as improving model comprehensibility even when development is centralized.

Warfare simulation has characteristics that make it a likely candidate for standard architectures and interfaces.[13] Consider the representation used by SIMNET. The intention is to facilitate "natural" interaction between software modules by representing combat at the level of vehicles using an object-oriented representation. This approach does avoid many problems associated with more abstract representations. Nonetheless, no approach, including this one, can completely avoid modeling problems associated with distributed development. A system designed to capture weapon-on-weapon interactions might require revisions to existing objects to add in the effects of electromagnetic pulse or other unanticipated area weapons. Further, the need to add objects at other levels of resolution (such as semi-automated forces) may always arise in general-purpose systems.

It is generally true that mapping between representations at different levels of resolution requires that specific modeling assumptions be made. Consequently, distributed systems involving models at differing levels of aggregation will be challenging to validate, and will typically be valid only for limited ranges of state. (Different states may require different aggregation and disaggregation functions.) It is im-

[13]Warfare simulation has a strong underlying conceptual model created by military doctrine. Thus, simulation models that differ widely in implementation may have numerous parallels (i.e., they all distinguish between defensive counter-air, offensive counter-air, close air support, and deep strikes). A standard architecture would be a formalization (and extension) of the common conceptual model. (An example of a model architecture that has been successfully used for multiple models may be found in Allen and Wilson, 1987.)

portant not to underestimate the modeling difficulties inherent in the more advanced DWS concepts.

STANDARDS FOR MODEL DEVELOPMENT

Reliable integration of separately developed submodels requires agreed upon structures that constrain the separate development efforts. Such agreed-upon structures would constitute standard specifications for warfare simulation models. Standards would do more than facilitate distributed development in that they could enhance the correctness and understandability of warfare simulation models in general. Devising and promulgating such standards is not an easy task, but the payoff is large. Partial success would be worthwhile, and there is a wealth of software engineering experience that has been little exploited by the modeling community.[14] At least three types of standards could prove useful:

- **Standard Conceptual Models.** Submodels sharing an overall conceptual model will be much easier to integrate than would otherwise be true. Models generally will be easier to understand, maintain, and validate if there is an explicit conceptual model created prior to implementation. A conceptual model would cover such details as model partitioning, submodel interfaces, and modeling assumptions. Developing an adequate standard for expressing such conceptual models will require experimentation, but the technical literature provides possible approaches (Berry, 1984; Berry and Berry, 1984; Britton et al., 1981; Hansen, 1973; Hill and van Cleemput, 1979; Parnas, 1979).
- **Standard Software Practices.** Further benefits are possible by coordinating distributed implementation efforts. Coordination can be achieved through standard tools, standard languages, graphics standards, and explicit model interfaces (which could be expressed as Ada package specifications [Berry, 1984] or standard data dictionaries).
- **Standards for Model Descriptions.** Aside from technical standards, the community also needs a standard format for readable (by humans) descriptions of the inner workings, modeling assumptions, and underlying philosophy of models. Such a standard

[14]Much of the progress in software engineering since the early 1970s has not had much impact on the building of military simulations. Adoption of modern software practices would be beneficial on its own merits and is critical if distributed simulation is to be a viable long-term option.

> would promote effective documentation, support model revision and interfacing, and ease the problems of understanding different combat simulations. At present there is a vague community ethic that one should document one's work. However, the lack of specification of what such documentation should look like undermines the ability to ensure that adequate documentation is generated or to compare documentation for different models. In addition to easing the interfacing and maintenance of models, a standard format for documentation could enhance the quality of models by easing the process of peer review. Peer review of models seldom occurs at present, a peculiar state of affairs given the tremendous importance of warfare simulation modeling.[15]

The development of standards of this sort must be regarded as a research issue with large potential benefits.

No matter what framework is used for the definition of a standard, there will be a need to gain experience from trial efforts before committing to any particular proposal. Too aggressive a standard, adopted too soon, can be worse than no standard at all. The worst case would be a standard devised by a committee, imposed bureaucratically, that handcuffs implementors who know better and severely impairs the quality of models. To avoid foolish, immature standards, it is necessary to adopt an incremental approach. Early standards should be modest, and be subjected to testing and redesign before they are extended to become more comprehensive.

Once a standard has been designed and approved, its use must be promoted and in some cases imposed. It must be anticipated that there will be resistance by members of the technical community who perceive the imposition of standards as a loss of freedom. However, as with any standardization effort, if a well-designed standard begins to be accepted, the benefits from using the standard far outweigh the constraints it imposes. Immature standards must not be imposed on modelers and adequate experimentation with standards must be supported, so that a mature standard can evolve.

There are other means by which a standard can be promoted other than bureaucratic fiat. An interesting possibility is the use of simulation shells. A simulation shell provides a software toolkit with tools

---

[15]It seems likely that this problem is due as much to political factors as technical ones. Here, as with many problems mentioned in this report, the potential of distributed simulation to force improvements in combat modeling may be one of its greatest virtues.

that ease the simulation-building process. The shell could also encourage the use of modeling standards. By incorporating the standards in a toolkit, the shell by itself can be a *de facto* standard. If the shell is useful on its own merits, it could promote a standard by making compliance automatic.

Alternatively, a standard could be agreed upon by a small consortium of interested agencies. Should their approach be successful, participation would increase as others adopt the standard. As standards gain popularity, the benefits of adherence grow as well. This can result in a snowball effect once the first wave of converts is won over.

QUALITY OF MODELS

Distributed simulation can increase the effect of warfare simulation, but the value of warfare simulation (whether distributed or not) is critically dependent on the quality of the underlying simulation models. There are numerous problems with existing simulation models and with the techniques of combat modeling generally. Problems include:

- Inadequate verification and validation of models,
- Insufficient sensitivity analysis,
- Poor comprehensibility of model runs,
- Poor visibility of modeling details and underlying assumptions,
- Unsatisfactory representation of difficult phenomena (e.g., frictional effects in warfare), and
- Poor understanding of issues in variable resolution modeling.

These problems must be addressed if a broad range of distributed warfare simulations is to be developed and understood. In fact, the quality of models is fundamental whether simulation is distributed or not. It can be argued that focusing on distributed warfare simulation is missing the central point, which is a general need to improve the quality of warfare simulation in a variety of ways. However, if distributed warfare simulation proves to be a goal around which significant support can be marshaled, it can serve as a rallying point for addressing broader problems that affect warfare simulations.

4. INCREMENTAL IMPLEMENTATION

Perhaps one day there will be a comprehensive distributed warfare system of global scope, supporting continuous exercises of personnel from all the services, on a variety of simulated weapons, both real and proposed. Should that day come, it is certain that the global DWS system will not have been constructed in one piece and left unchanged. Incremental construction and evolving capabilities are both more realistic, and more desirable. The next question is how to incrementally develop a capability for distributed warfare simulation. There are several reasons why a long-term distributed warfare simulation capability must be acquired incrementally:

- Some technologies may become relevant but are as yet immature,
- Experience in distributed applications will redefine requirements, and
- Fiscal constraints may dictate spreading the investment over a number of years.

The incremental implementation of the capability has associated with it two principal issues:

- Providing adequate support for long-term development as well as nearer-term applications, and
- Promoting coordination between research and development efforts targeted at various time horizons.

If most of the potential benefits discussed in Section 2 are to be realized, it is important that long-term research and development be pursued along with near-term applications. Incremental development of the long-term capability can be achieved through progressively more challenging applications, with the products of research and prototype development feeding in smoothly as they become ready.

Ideally, the various efforts should be pursued in a coordinated fashion—making longer-term development relevant to the applications of interest and pursuing near-term objectives in a fashion that maximizes contribution to long-term goals.

Regarding the latter point, *it is especially important not to freeze in bad design choices through sunk costs*. Long-term design issues must be confronted in the development of near-term applications. Unless near-term efforts are guided by long-term intentions, much of the capital invested in near-term projects will not advance long-term goals. Thus, the design of longer-term systems should be pursued in parallel to near-term development.

This design or vision should be shared across as large a portion of the community of developers as possible. Lack of shared design can lead to a "bottom-up" design process, in which various pieces of the ultimate system are developed separately and must later be integrated. This integration step can be difficult if the various pieces have not been designed to work together.

In the remainder of this section, we describe various aspects of a hypothetical long-term effort to develop a capability for distributed warfare simulation. This effort consists of the following activities:

- Development of specifications and high-level design,
- Near-term applications,
- Development of engineering prototypes,
- Research to support long-term needs, and
- Community building activities.

In the ideal program, all of these components would be strongly interconnected, with research efforts being guided by the problems experienced in applications and with new technology being incorporated whenever it is ready. Achieving this sort of coordination will require better definition of managerial responsibility for the general effort. Along with strong management, communication and understanding must be promoted among the community of developers.

SPECIFICATIONS, HIGH-LEVEL DESIGNS, AND MEASURES OF EFFECTIVENESS

If the efforts of the developmental community are to be coordinated, a central need is to build a vision of the system that we would like to construct. This vision must reflect both users' needs and technological capabilities. Several activities should be promoted in this regard:

- **Functional specifications.** Efforts to examine design options for future systems are greatly impaired by the lack of concrete specifications of users' needs. Ideally, what sort of exercises would be run, how often, and with what training audience? How are these needs prioritized? These questions need to be answered to assess cost/benefit ratios and to evaluate possible designs.

- **Long-term plan.** A plan to incrementally meet the needs of the user community will take the form of a series of target systems, each incrementally more advanced.
- **High-level design.** Each target system can be specified by means of a high-level design that would identify major system components and specify the characteristics those components would have. The design of the eventual system will be much more detailed and possibly revised in some places. Nonetheless, an existing high-level design for future systems can influence the detailed design of near-term systems, orient research efforts, and provide specifications for prototype development efforts.
- **Standards.** Because standardization is difficult, although crucial to many long-term interests, the incremental development and adoption of standards should be encouraged. Initial standards will be experimental and limited. Experience gained from initial efforts should allow later versions to be more comprehensive.
- **Measures of effectiveness (MOEs).** For experience gained from earlier implementations to be incorporated in later generations it is useful to develop means for assessing where the system is successful and where it fails to provide desired features. (In a distributed development effort, it may also be necessary to decide *a priori* who will make this assessment.)

APPLICATIONS

In the process of pursuing applications, opportunities exist to support the other phases of the general DWS effort:

- Where possible, make all development work on applications consistent with longer-term designs. The importance of this is directly proportional to the amount of resources being invested. If a large effort is being undertaken, it should contribute to both long-term and short-term goals.
- Use proposed standards as they become available.
- Provide input to longer-term efforts by means of lessons learned. For example, the interface standards development could be facilitated by documenting problems that arise in the process of integrating separately developed models.

- Encourage coordination among various agencies (as well as pushing the state of the art) through an aggressive joint application.

DEVELOPMENT OF ENGINEERING PROTOTYPES

Technology transfer is difficult in systems supporting ongoing games, since major modifications negatively affect reliability and jeopardizing important exercises constitutes an unacceptable risk. These problems can be avoided by developing engineering prototypes containing advanced features. Longer time lines associated with development efforts allow for good design standards (i.e., flexibility, understandability, modifiability). Testing and reimplementation of these prototypes permits "working the bugs out" prior to releasing the system for general use. Because these prototype systems address real problems, not scaled-down pieces such as may be required for research, they allow technologies that are near maturity to be tested and refined.

If DWS systems are built using a layered architecture, it will seldom be necessary to build a prototype system entirely from scratch. Instead, lower layers of the existing architecture can be used in a prototype that is modifying higher levels. Thus, if early implementation incorporates a design for a telecommunications architecture with sufficient upward compatibility, there eventually could arise a general DWS network, partitioned into segments in which games for different purposes (including experimental development) could be pursued.

Such a network would initially take the form of a distributed testbed facility. Such a facility could be implemented around existing institutions by providing communications links, support software, and standards that would allow all of the nodes (participating institutions) to work together. A testbed would support communications between developers, distributed development of prototype systems, and distributed testing and evaluation of prototype systems. An alternative approach would be to augment the training capabilities of operational combat data systems (as the Navy has considered doing with AEGIS).

COMMUNITY BUILDING

The use of simulation models for analysis and gaming of military conflicts is a technology in which the community of expertise is poorly coordinated and has few means for communication and discussion. Problems that are serious in any case must be resolved if progress is to be made toward DWS. Options for promoting communication and coordination include:

- **Scientific committees to specify warfare simulation standards.** Even if time is required to produce mature standards, committees can promote thinking about the problem and serve as a vehicle for communication among representatives of various segments of the community. Such committees are prone to be political and particular standards efforts may fail. Initial efforts should be viewed as experimental, and successful standards may require multiple iterations.
- **Workshops and conferences.** The workshop on distributed warfare simulation demonstrated the enormous need the community has for forums in which to exchange views about the various problems confronting it. Support for additional conferences of this sort, including some aimed at a wider array of topics, would serve a good purpose.
- **Journals and textbooks.** This field is sadly lacking in channels to communicate results. We need professional journals and textbooks for combat modeling.
- **Peer review.** Opportunities for peer review of models need to be created. At present there are few opportunities for professional appraisal of the technical merits of new work.
- **Executive working groups.** Numerous agencies may support the development of distributed warfare simulations. Executive committees could devise means of promoting coordination, and thus avoid conflict, duplication of effort, or incompatibilities of systems that may eventually be merged.

BIBLIOGRAPHY

Adams, Richard G., "Networking Gurus Take First Step Toward Simulator Interoperability," *Armed Forces Journal International*, November 1991, pp. 49–51.

Adelman, Leonard, Deborah A. Zirk, Paul E. Lehner, Raymond J. Moffett, and Richard Hall, "Distributed Tactical Decisionmaking: Conceptual Framework and Empirical Results," *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. SMC-16, No. 6, 1986, pp. 794–805.

Advanced Military Computing, April 11, 1988.

Allen, Patrick D., and Barry A. Wilson, *Secondary Land Theater Model*, RAND, N-2625-NA, July 1987.

American National Standards Institute, Draft Proposal ISO/DP 97/SC16/537 Rev., "Data Processing—Open Systems Interconnection Basic Reference Model," New York, March 31, 1981.

Bankes, Steven, and Carl Builder, with Robert Anderson, Richard Bitzinger, Hugh DeSantis, Constance Greaser, Peter Jacobson, Dana Johnson, Richard Leghorn, Jeff Marquis, Dean Millot, David Ronfeldt, and Norman Shapiro, *Seizing the Moment: Harnessing the Information Technologies*, RAND, N-3336-RC, forthcoming.

Bastani, Farokh, Wael Hilal, and S. Sitharama Iyengar, "Efficient Abstract Data Type Components for Distributed and Parallel Systems," *IEEE Computer*, Vol. 20, No. 10, October 1987, pp. 33–44.

Beckman, B., et al., "Distributed Simulation and Time Warp, Part I: Design of Colliding Pucks," in Brian Unger and David Jefferson (eds.), *Distributed Simulation*, Vol. 19, No. 3, Simulation Councils, Inc., San Diego, California, 1988.

Berets, James C., Ronald A. Mucci, and Richard E. Schantz, "Cronus: A Testbed for Developing Distributed Systems," IEEE Military Communications Conference, 1985.

Berry, Clifton F., Jr., "Recreating History: The Battle of 73 Easting," *National Defense*, November 1991, pp. 52–56.

Berry, Dan M., "On the Use of Ada as a Module Interface Description," *Proceedings of the Hawaii International Conference on System Sciences*, January 1984.

Berry, D. M., and O. Berry, "The Programmer-Client Interaction in Arriving at Program Specifications: Guidelines and Linguistic Requirements," *Proceedings of the IFIP TC2 Working Conference on System Sciences*, January 1984.

Breibart, Y., A. Silberschatz, and G. Thompson, "An Update Mechanism for Multibase Systems," *Data Engineering*, Vol. 10, No. 3, September 1987, pp. 12–18.

Britton, Kathryn Heninger, R. Alan Parker, and David L. Parnas, "A Procedure for Designing Abstract Interfaces for Device Interface Modules," *Proceedings of the 5th International Software Engineering Conference*, IEEE Computer Society Press, March 1981, pp. 195–204.

Brooks, Frederick P., *The Mythical Man-Month*, Addison-Wesley, Reading, Massachusetts, 1982.

Builder, Carl, and Steven Bankes, *The Etiology of European Change*, RAND, P-7693, 1990.

Ceri, S., and G. Pelagatti, *Distributed Databases, Principles and Systems*, McGraw-Hill, New York, 1984.

Edmond, Winston, Steven Blumenthal, Andres Echenique, Steven Storch, Tom Calederwood, and Tom Rees, *The Butterfly Satellite IMP for the Wideband Packet Satellite Network*, BBN Laboratories Inc., Cambridge, Massachusetts, 1986.

Fox, G., M. Johnson, G. Lyzenga, S. Otto, J. Salmon, and D. Walker, *Solving Problems on Concurrent Processors: General Techniques and Regular Problems*, Prentice-Hall, Englewood Cliffs, New Jersey, 1988.

Hansen, Per Brinch, *Operating Systems Principles*, Prentice Hall, Inc., Englewood Cliffs, New Jersey, 1973.

Hill, D., and W. van Cleemput, "SABLE: A Tool for Generating Structured, Multi-Level Simulations," *Proceedings of the 16th Design Automation Conference*, June 1979, pp. 272–279.

Jefferson, D., and H. Sowizral, *Fast Concurrent Simulation Using the Time Warp Mechanism, Part I: Local Control*, RAND, N-1906-AF, December 1982.

Kahan, James P., D. Robert Worley, Suzanne M. Holroyd, Leland C. Pleger, and Cathleen Stasz, *Implementing the Battle Command Training Program*, RAND, R-3816-A, August 1989.

Landers, T., and R. L. Rosenberg, "An Overview of MULTIBASE," in H. J. Schneider (ed.), *Distributed Data Bases*, North Holland Publishing Company, Amsterdam, September 1982.

Lindsay, B. G., "A Retrospective of R*: A Distributed Database System," *Proceedings of the IEEE*, Vol. 75, No. 5, May 1987, pp. 668–673.

LOCUS Computing Corporation, *LOCUS Distributed UNIX: A Comparison with UNIX*, May 1984.

Lyons, Robert E., and John H. Cushman, "Toward a NATO Capability for Distributed Warfare Simulation," NATO Workshop on Political-Military Decision Making, 1987.

Marti, J., "RISE: The RAND Integrated Simulation Environment," in B. Unger and D. Jefferson (eds.), *Distributed Simulation*, Vol. 19, No. 3, 1988.

National Security Industrial Association, Command, Control, Communications and Intelligence Committee, *Simulation and Modeling Study, Phase II, Final Report*, February 1987.

Parnas, David L., "Designing Software for Ease of Extension and Contraction," *IEEE Transactions on Software Engineering*, SE-5(2), March 1979, pp. 128–137.

Pope, Arthur R., *The SIMNET Network and Protocols*, BBN Systems and Technologies Corporation, Report No. 7102, Cambridge, Massachusetts, 1989.

Pope, Arthur R., and Duncan C. Miller, *The SIMNET Communications Protocol for Distributed Simulation*, BBN Laboratories, Cambridge, Massachusetts, 1987.

Popek, G., and B. Walker, *The LOCUS Distributed Operating System*, MIT Press, Cambridge, Massachusetts, 1987.

Pu, C., "Superdatabases: Transactions Across Database Boundaries," *Data Engineering*, Vol. 10, No. 3, September 1987, pp. 19–25.

Rehmus, Paul, *Overview of the Warrior Preparation Center's Joint Warrior Exercises*, RAND, P-7305, June 1987.

Sarin, S., "Letter to the Editor," *Data Engineering*, Vol. 10, No. 3, September 1987, pp. 3–5.

Scott, Michael L., *The Interface Between Distributed Operation System and High-Level Programming Language*, University of Rochester, Department of Computer Science, Technical Report TR-182, 1986a.

Scott, Michael L., *Language Support for Loosely-Coupled Distributed Programs*, University of Rochester, Department of Computer Science, Technical Report TR-183, 1986b.

Shatz, Sol M., and Jia-Ping Wang, "Introduction to Distributed-Software Engineering," *IEEE Computer*, Vol. 20, No. 10, October 1987, pp. 23–32.

Spector, Alfred Z., "Performing Remote Operations Efficiently on a Local Computer Network," *Proceedings of the Eighth Symposium on Operating Systems Principles*, Vol. 15, No. 5, Special Interest Group on Operating Systems, Association for Computing Machinery, 1981.

Stonebreaker, M., "Concurrency control and consistency of multiple copies of data in distributed INGRES," *IEEE Transactions on Software Engineering*, SE-5(3), May 1979, pp. 188–194.

Tanenbaum, Andrew S., *Computer Networks*, Prentice-Hall, Englewood Cliffs, New Jersey, 1981a.

Tanenbaum, Andrew S., "Network Protocols," *Computing Surveys*, Vol. 13, No. 4, December 1981b.

Templeton, M., D. Brill, S. K. Dao, E. Lund, P. Ward, A.L.P. Chen, and R. MacGregor, "Mermaid—A Front-End to Distributed Heterogeneou Databases," *Proceedings of the IEEE*, Vol. 75, No. 5, May 1987, pp. 695–708.

Unger, Brian, and David Jefferson, *Distributed Simulation*, Vol. 19, No. 3, Simulation Councils, Inc., San Diego, California, 1988.

Wah, A Ben, and Jie-Yong Juang, "Resource Scheduling for Local Computer Systems with a Multiaccess Network," *IEEE Trans. Computers*, Vol. C-34, No. 12, December 1985, pp. 1144–1157.

Ward, A. A., "TRIX: A Network-Oriented Operating System," *IEEE Computer Society International Conference*, IEEE, Long Beach, California, Spring 1980, pp. 344–349.

Wever, Peter, Eric Lang, and C. S. Smyth, *SIMNET Database Interchange Specification*, BBN Systems and Technologies, Cambridge, Massachusetts, Report No. 7108, 1989.

Wulf, W. A., et al., "Hydra: The Kernel of a Multiprocessor Operating System," *Communications of the ACM*, Vol. 17, No. 6, 1974, pp. 337–345.