A Methodology for Using Intelligent Agents to Apply Simulation Technologies to the Mission Operational Environment

John R. Surdu and Udo W. Pooch
Texas A&M University, Department of Computer Science, College Station, TX, USA
{surdu | pooch}@cs.tamu.edu

Keywords: Simulation, Operation Monitoring, Intelligent Agents, Command and Control

ABSTRACT: This paper proposes a methodology for applying simulation technology to the monitoring and control of operations, focusing on the roles of intelligent agents. These agents monitor the course of the real operation and compare it with a near-real-time simulation of that operation. When these agents detect significant differences between the planned operation and the real operation they explore the discrepancies to determine if they impact on the desired outcome of the operation. When necessary, these agents launch additional simulations or other tools to make this determination. In cases in which the success of the plan is threatened, the agents advise the decision-maker. This paper focuses on the interactions between the agents and how they collaborate with each other in a hierarchy.

1. INTRODUCTION

A large number of tasks performed by commanders and staffs can be facilitated during operations through the application of simulation technologies. Traditionally the focus of simulation in the Department of Defense (DoD) has been on analysis and training. Simulations designed to facilitate course of action (COA) development and analysis, rehearsal, and operations monitoring can greatly enhance the effectiveness of staffs and commanders.

The Army Modeling and Simulation Office (AMSO) has identified five modeling and simulation technology voids for the Army After Next[†] (AMSO 1998). This list includes automated decision aids, COA tools, and tactical information aids. This methodology, originally proposed by Surdu and Pooch (Surdu 1998), intends to fill these three technology voids.

This paper does not elaborate on the uses of simulation for COA development and analysis or the use of simulations for rehearsals. (Surdu, Haines, and Pooch (Surdu 1999a) discussed these uses of simulation.) Instead it is assumed that these processes have been completed and that the operation is about to commence. First, this paper discusses the use of simulation during an ongoing operation as a tool to help the commander and his staff track the progress of the operation and anticipate future problems. Then it proposes a methodology in which simulation technologies support commanders and staffs during actual operations. Finally it discusses the technical issues arising from this methodology. In particular, the focus is on the organization, collaboration, and dynamic instantiation of rational, utility-based agents to conduct the analysis and take advisory actions.

2. SIMULATION *DURING* OPERATIONS

While the various services and civilian organizations have different command and control processes, there is a common thread among them: planning, rehearsal, execution, and after action review. Simulation technology can be applied in each of these phases.

During the planning phase, staffs develop courses of action (COAs). With an *operationally-focused* simulation, the staff enters enemy and friendly courses of action and then simulates them. The results of numerous simulation experiments provide feedback to the decision-maker in choosing one COA over the others. Once a COA has been chosen, it is developed into a full plan and that plan is rehearsed. The simulation facilitates this detailed planning and rehearsal.

After the plan has been chosen, refined, and rehearsed, and the operation commences, the methodology we propose can be used to monitor the progress of the simulated plan and the real operation. Operations Monitors compare the progress of the real plan against the simulation of that plan. When significant deviations from the plan occur, the Operations Monitors launch tools that explore the impact of these deviations. Finally, the commander is advised if the Operations Monitors determine that the success of the plan is in jeopardy.

The Defense Advanced Projects Research Agency (DARPA) has recognized the importance of simulation in command and control activities. In its concept briefing for the Command Post of the Future (CPoF) project, DARPA lists several tools that will provide input to the Battlespace Reasoning Manager. Among these are "Planning and Analysis Applications" and "3D Models and Simulations." In another portion of the briefing, DARPA notes that "Battlespace Reasoning, Analysis, and Simulation" assist the commander's perception and understanding of the Battlespace (DARPA 1998).

[†] Army After Next is the vision of the structure and doctrine of the U.S. Army after 2010. It is an ongoing process.

3. WHY NOT USE TRAINING SIMULATIONS?

The military community has developed many simulations for training and analysis, such as Corps Battle Simulation (CBS) and JANUS. While many of these are excellent products, most are unsuitable for use during an operation for a number of reasons, including large pre-exercise preparation, large numbers of required participants, and large numbers of required workstations.

Surdu, Haines, and Pooch (Surdu 1999a) enumerated the desirable capabilities for an operationally-focused simulation for use during operations. They include:

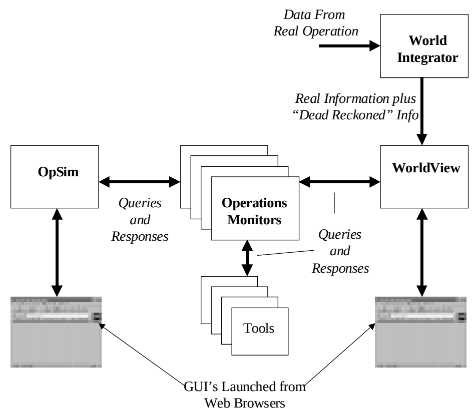
- The simulation must be runable from a single workstation by a single user.
- The simulation must be runable on low-cost, open systems, multi-platform environments.
- The simulation must be capable of running in multiples of wall-clock time (i.e., real time and much faster than real time).
- The simulation must be able to receive and answer queries from external agents.
- If needed, multiple simulations should be capable of operating together.
- The simulation should be based on an aggregate-level model.

In addition, the simulation must be capable of producing information that stimulates all seven Battlefield Operating Systems. Surdu, Haines, and Pooch implemented a simulation that meets these requirements; however, this methodology does not rely on that simulation. Any simulation that meets these requirements could be made to support this methodology.

One government-developed simulation that does not currently have all the properties described, but could be so modified, is ModSAF[‡]. While ModSAF and its proposed follow-on product, OneSAF, are entity-level simulations, their Distributed Interactive Simulation (DIS) and Persistent Object Protocol (POP) protocols could be wrapped in an "agent," which would manage the receipt and answer of subscriptions and queries. ModSAF is not inherently *cross-platform*, but it has been ported to a variety of platforms, and the GUI (which communicates with the simulation via UDP/IP messages) could be rewritten in a language like TCL/TK or Java to provide this capability.

4. PROPOSED METHODOLOGY

A proposed methodology is summarized in Figure 1. The methodology involves the interactions of a number of packages and tools, including the operationally focused simulation discussed briefly in Section III, intelligent agents, combat attrition models, path-planning algorithms, etc. Each of the various components of the methodology is discussed below.



OpSim: The operationally focused simulation runs in near real time, tracking the predicted progress of the plan. The progress of this simulation can be monitored from the web-based GUI. The Operations Monitors (OMs), discussed below, register interest in various entities and events with the simulation, and they query the simulation directly for information.

Operations Monitors (OMs): The OMs are the heart of this methodology. They perform two important functions. First, they can take information from WorldView and update the state of entities in OpSim, seamlessly re-synchronizing the simulation to the real world. Second and more importantly, they monitor the progress of the simulation, comparing it with the progress of the real operation. When the OMs discover significant deviations between the real world (WorldView) and the simulated world (OpSim), events referred to as *Potential Points of Departure* (PPDs), they launch one or more tools to explore the ramifications of these deviations.

It is important to note that OMs do not take actions with regard to the plan; rather, they explore the ramifications of differences between the real operation and the planned operation. The job of OMs is to help human decision-makers manage information (Maes 1994a, 1994b). OMs should be considered part of the team, not a replacement for decision-makers (Hayes-Roth 1992). OMs make some judgement about the seriousness of any differences and issue advisories to the decision-makers.

OMs must be proactive. It is not sufficient for them to inform decision-makers that some timetable has been broken. The OMs must look ahead to future events

‡ Modular Semi-Automated Forces (ModSAF) was originally built by Loral Advanced Distributed Simulations. It is under configuration control by Simulation Training and Instrumentation Command (STRICOM), Orlando, FL.

and inform decision-makers if some goal is unlikely to be met. For instance, if some future even requires three of five preconditions be met, the OM must determine whether these preconditions are likely and assess the probability that the eventual goal can be accomplished. When this probability drops below some threshold, the OM must inform the decision-maker.

OMs are implemented as rational, utility-based agents. There are a number of useful definitions of what is an agent. Franklin and Graesser (Franklin 1997) proposed a list of characteristics that characterize a piece of software as an agent: autonomous, reactive, goal-oriented, persistent, communicative, adaptive, mobile, and flexible. According to Franklin and Graesser, software which has the first four characteristics is an agent, and software which exhibits some of the other attributes in the list fall into more specialized categories of agents. We use this method in defining OMs within this methodology as agents.

Each OM is interested in only a narrow domain. By focusing each OM on a very narrow domain, the problem of building intelligence into these agents becomes more tractable. A promising approach for the implementation of these agents is hybrid machine learning, such as KBANN or EITHER (Mitchell 1997, Ourstan 1994).

PPDs are not always held in fixed, global knowledge bases. Instead they are domain specific. Each OM has a knowledge base that it uses to analyze the discrepancies between the real operation and the simulated plan. Because of the inherent uncertainty in the knowledge associated with the domains, non-crisp reasoning (or soft computing) often is required of these agents. PPDs may be represented in a variety of ways, depending on the purpose of the OM. For example, the PPDs may reside in a fixed knowledge base (which may be in the form of rules) (Giarratano 1989). PPDs for other OMs may be in the form of some refinable domain theory (Mitchell 1997). PPDs may also be represented in a fuzzy inference engine like the one developed by the National Oceanographic and Atmospheric Agency for their Weather Scenario Generator (Kihn 1999). This engine is used to search through weather data based on linguistic descriptions of the information required.

WorldView: The WorldView module is a representation of the real operation. In order to simplify the task of the OMs, the representation of the state of the real operation and the simulated plan should be as similar as possible. WorldView receives information about the state of the real operation through a series of APIs. It then transforms this information into a form that the OMs can easily interpret.

WorldIntegrator: WorldIntegrator has the onerous task of monitoring the real operation, processing that information, and passing it to WorldView. In some systems, such as the Global Command and Control System (GCCS), this may involve querying a database. In other systems, this may require "eavesdropping" on the network. The reason for this intermediate step is that in real operations, reports on some entities may be intermittent. It is the job of WorldIntegrator to "dead reckon" these intermittent reports and pass them into WorldView. Clearly, when an entity has been "dead reckoned," this must be reflected in the information that WorldView gives to the OMs.

The issue of WorldIntegrator and WorldView involves sensor, data, and information fusion. WorldIntegrator must determine when an entity has been unconfirmed long enough that its actions must be dead reckoned. When some sensor reports a similar unit, the WorldIntegrator must determine whether this is merely the lost unit reappearing or a different unit. These and other issues regarding sensor, data, and information fusion are open research issues.

Tools: We do not attempt to enumerate all possible, useful tools. Instead, we give examples of tools and how the OMs might use them. For example, if the Enemy OM noted, based on information from WorldView, that there are two enemy, mechanized battalions in the area of operations rather than the one assumed during COA analysis, it must take some action. The OM might call a combat attrition model to determine the difference in expected losses, or the OM might merely apply a closed form solution to Lanchester equations to get a quick estimate of expected losses. If it appears that this difference will adversely affect the plan, the Enemy OM will notify the decision-maker.

Similarly, if the Mission and Time OMs note, for example, that a ground unit had missed an important phase line by forty-five minutes, they might launch another simulation to explore how this would effect other units. If the effect is minimal, the OM might recommend to the commander that the overall time line be shifted forty-five minutes to resynchronize the simulation.

There is a great deal of interaction between the OMs and OpSim and between the OMs and WorldView. This interaction is conducted through a message-passing protocol. There are two kinds of request messages: individual queries and registration of interest (subscriptions). An OM, for example, might send a query to WorldView and OpSim about the status of a particular unit. This is done as an individual query. An OM might also register interest in certain information. For instance, the Troops OM, might register for periodic updates of the strengths of units. Registration of interest is preferred, since in an ongoing basis, it requires roughly half the number of messages as individual queries. OpSim launches a separate thread to handle each of these registrations.

In order for the OMs to adequately compare the real operation with the simulated operation, the two representations must be "close." An axiom in military planning is that no plan survives the first rifle shot. Once the operations commence, the plan will certainly diverge, at least to some extent, from the real operation. The job of the collection of OMs is to identify when this divergence has become so great that the success of the operation is in jeopardy. The OMs report this concern to the decision-maker (probably with some degree of certainty attached to this conclusion).

Once the decision-maker has been notified that the currently running simulation no longer accurately reflects the state of the actual operation, the simulation should be updated. If the simulated plan continues to diverge from the real operation over time they will become almost completely unrelated. Any analysis the OMs would perform at that point would be meaningless. This updating also allows OpSim to better predict the state of the operation in the future. The problem, however, is to define a synchronization mechanism which is feasible and adaptive. An approach to solving this problem was outlined by Surdu and Pooch (Surdu 1999b).

This updating of the simulation is crucial to the success of the methodology proposed. First, it keeps the simulation "close" to the current operation, allowing the OMs to make valid comparisons. Second, it periodically synchronizes the simulation to the current state of the operation to facility projecting (simulating) the current situation into the future. Third, it allows the simulation to adapt itself so that future predictions will be more accurate.

OMs do not make tactical decisions. They explore differences and report findings. The autonomy of the OM lies in its ability to decide when and if to launch other tools. As noted in the DARPA CPoF concept, battlefield visualization tools must be decision-centered. Among other things, this means that these visualization systems "show decision-relevant details, highlight relevant changes, anomalies, [and] exceptions, and portray uncertainties" (DARPA 1998). These are exactly the pieces of information that our proposed methodology is designed to provide. Visualization is not a tool to show the battlefield in a unique way; visualization is a process that occurs within the heads of the commander and his staff (U.S. Army 1996). Our proposed methodology provides additional support for this process.

5. HOW OMs ARE CREATED DYNAMICALLY

As stated earlier, OMs focus on a narrow domain. This makes their design and implementation more tractable. When the system is first launched, a manager OM creates the first layer of OMs in the hierarchy. The overall manager is responsible for synthesizing the reports of the agents below it in the hierarchy. The first layer of OMs in the hierarchy compare the current situation with the plan, each looking at the operation from a particular, narrow perspective. One such taxonomy for OMs in this first layer is the use of the Battlefield Operating Systems: maneuver; fire support; air defense; command and control; intelligence; mobility, counter-mobility, and survivability; and combat service support (logistics and personnel).

These OMs in the first level of the hierarchy have a number of tools (and additional agents) available to them to perform their analysis. Each OM uses a rule-based expert system to make inferences, determine whether to launch additional tools to help with analysis, and decide what actions to take. There are basically three types of rules: those which dispatch other agents or tools, those which report information to other agents, and those which take other actions (e.g., advising the human of problems or updating the simulation as discussed earlier). Some of these rules are domain dependent; they are related to the particular focus of the OM.

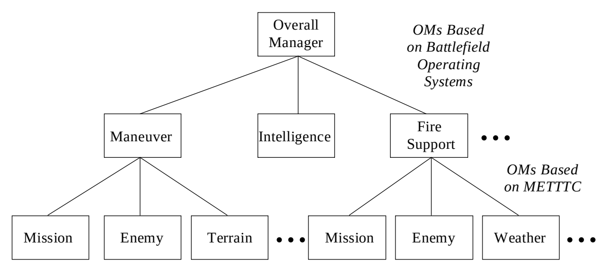


Figure 2: A Possible, Partially-Expanded Hierarchy of OMs

Other rules are domain independent; they are related to general issues, such as the resources needed by a particular candidate tool. Domain independent rules take into account RAM usage, time complexity, bandwidth, etc. (as shown in Table 1). These domain independent rules are important, because they help insure that the system does not grind to a halt during times of peak activity during the operations. For instance, the eventual system might have two different path planning algorithms that could be used to determine the impact of rerouting

Table 1: Sample Structure of Tool Classifiactions Used by anOM in Computing the Utility of a Tool

SYMPTOM 1
  CAUSE 1, P( ), C( )
    TOOL A MEMORY (N), O (N), BANDWIDTH (N)…
    TOOL B MEMORY (N), O (N), BANDWIDTH (N)…
    TOOL C MEMORY (N), O (N), BANDWIDTH (N)…
  CAUSE 2, P( ), C( )
    TOOL B MEMORY (N), O (N), BANDWIDTH (N)…
    TOOL D MEMORY (N), O (N), BANDWIDTH (N)…
SYMPTOM 2…

logistics. If one algorithm has a smaller time complexity but is somewhat less accurate than the other algorithm, the OM might choose to use this algorithm if the host on which it is running is already very busy. As mentioned earlier, OMs are utility-based agents, and these domain independent rules provide information used to compute the utility of a particular action.

A possible taxonomy for agents in this second layer of OMs might be the Army's METTTC mnemonic (Mission, Enemy, Time, Troops, Terrain and Weather, Civilian Impact). Under this taxonomy, one OM would be looking for differences in the size, strength, and/or composition of the enemy. Another might be looking at effects of terrain and weather.

One possible, partial expansion of an OM hierarchy is shown in Figure 2. Note that an instantiation of a particular class of OM can exist at multiple points in the hierarchy. The only restriction is that an OM cannot call a tool above it in the hierarchy to avoid circular dependencies. Since any OM can create any number of subordinate OMs to aid in its analysis based on the current situation and delete OMs that are no longer necessary, the hierarchy is dynamic.

This discussion of OMs has repeatedly referred to the use of rules. This is for ease of discussion. This methodology does not require or rely on any particular reasoning mechanism. OMs could use a variety of reasoning technologies, including crisp rule-bases, fuzzy rules bases, machine learning techniques, artificial neural networks, etc. The OM's domain and mission will probably dictate the appropriate reasoning mechanism. OMs must be capable of launching other tools as necessary, making inferences about the current situation, and taking actions as appropriate. This methodology does not specify the technology used to perform these tasks.

6. RELATED WORK

Wayne Davis at the University of Illinois at Urbana-Champaign has demonstrated the usefulness of what he terms on-line simulation for industrial applications (Davis 1998a). Davis' demonstrations include intelligent controllers for managing the production of products within a manufacturing system. Another interesting application is a Java applet which navigates a boat through a field of up to eight homing torpedoes to some objective (Davis 1998b). While a single boat on a flat, confined body of water is a simpler domain than opposing ground forces maneuvering over terrain, he has demonstrated the usefulness of on-line simulation to adversarial situations.

There is a great body of work related to Intelligent Agents research. Both Brooks, with his subsumption architecture (Brooks 1985 and 1991), and Minsky, in his Society of Mind (Minsky 1986), showed that interesting and complex behavior need not be the result of complex control systems. Coen showed that a hybrid of the subsumption architecture and traditional AI techniques (Coen 1999) can produce very complex behaviors in the MIT intelligent room project. In their approach, dubbed "Scatterbrain," a network of up to twenty, simple, interconnected agents control an intelligent room designed to facilitate new forms of collaboration between humans and computers. Coen's work does not include an explicit hierarchy as described in this paper, but the complex behaviors generated by simple, interacting agents point to the feasibility of our proposed approach.

7. FUTURE WORK

A prototype OpSim that allows subscription to information by OMs has been built (Surdu 1999a). It is an aggregate-level, discrete event simulation capable of near-real-time and faster operations. Future work on OpSim includes:

- Improving the query response capability of the simulation and permitting one-time queries in addition to subscriptions.
- Creating the ability for OpSim to run different plans at different speeds (i.e. in different threads) so that it could be running the current operation in near real time while running the plan (or a branch or sequel) as fast as possible to predict its outcome. This would also permit planning the future operation to be interleaved with monitoring the current operation rather than treating them as time-ordered, separate processes.
- Improving the statistics gathering and reporting capabilities of the simulation.
- Improving the ability to get information from the terrain database.
- Improving the API.
- Making the simulation compliant with various DoD protocols. As an aggregate-level simulation, it should be compliant with a protocol like the DoD Aggregate Level Simulation Protocol (ALSP), and the intent is that it eventually will be HLA compliant with the ALSP Federation (MITRE 1993).

While the bulk of this paper speaks in terms of military applications, we are actively exploring several civilian applications. One particularly promising application is fighting forest fires. The planning process is very similar, and the need for exploring the effects of changing situations is very important.

8. SUMMARY

This paper proposes a methodology for using simulations during an ongoing operation. This includes the use of an operationally focused simulation that runs in real-time, simulating the plan. This methodology also includes the use of intelligent agents, Operations Monitors, to compare the events in the real operation versus those in the plan. These agents query both the representations of the real operation and the simulation to find deviations from the plan. The agents then launch various tools to determine the effects of these deviations. If the effects are significant, the agents advise the commander and staff.

9. REFERENCES

AMSO 1998. Major Patrick J. Delany at the Army Modeling and Simulation Office Policy and Technology Working Group video teleconference, 28 October 1998.

Brooks, R. 1985. "A Robust Layered Control System for a Mobile Robot." AI Laboratory Memo 864. Massachusetts Institute of Technology, Cambridge, MA.

Brooks, R. 1991. "Intelligence without Representation." In Special Volume: Foundations of Artificial Intelligence, Artificial Intelligence, 47, 1-3.

Coen, M.H. 1997. "Building Brains for Rooms: Designing Distributed Software Agents." American Association for Artificial Intelligence.

DARPA 1998. Defense Advanced Research Projects Agency Command Post of the Future web site. URL: http://mole.dc.isx.com/cpof and http://www-code44.nosc.mil/cpof.

Davis 1998a. Davis, WJ. 1998. *Developing an Intelligent Controller for Integrated Manufacturing Resource Planning.* Manufacturing Systems Laboratory, University of Illinois at Urbana-Champaign, Urbana, IL, 61801 (Oct.).

Davis 1998b. Davis, WJ. 1998. Online Simulation for Torpedo Avoidance, URL: http://www-msl.ge.uiuc.edu/~brook/boat.

Franklin 1997. Franklin, S. and A. Graesser, "Is it an Agent, or Just a Program?: A Taxonomy for Autonomous Agents." In *Proceedings of the Workshop on Agent Theories, Architectures, and Languages: Intelligent Agents III*. (Berlin, Germany, 12 - 13 August 1997), vol. 1193 of *Lecture Notes in Artificial Intelligence*, 21-36. URL: http://www.msci.memphis.edu/ ~franklin/AgentProg.html.

Giarratano 1989. Giarratano, J. and G. Riley. 1989 *Expert Systems: Principles and Programming.* PWS-Kent Publishing Co., Boston, MA.

Hayes-Roth 1992. Hayes-Roth, B. et. al. 1992. "Guardian: A Prototype Intelligent Agent for Intensive Care Monitoring." *Journal of AI in Medicine: Special issue on Expert Systems for the Intensive Care Setting*, 165-185.

Kihn 1999. Surdu, E.A., et. Al "The Weather Scenario Generator." In *Proceedings of the 1999 International Conference on Web-Based Modeling and Simulation* (San Francisco, CA, 17-20 January 1999). SCS, San Diego, CA, 69-74.

Maes 1994a. Maes, Pattie. 1994. "Agents That Reduce Work and Information Overload," *Communications of the ACM*, vol. 37, no. 7 (Jul.), pp. 31-41.

Maes 1994b. Maes, Pattie. 1994. "Situated Agents Can Have Goals," *Designing Autonomous Agents: Theory and Practice from Biology to Engineering and Back*, Pattie Meas, ed. MIT Press, Cambridge, Massachusetts.

Minsky, M. 1986. *Society of Mind*. Simon and Schuster, New York, NY.

Mitchell 1997. Mitchell, T.M. 1997. *Machine Learning*. McGraw-Hill, Boston, MA.

MITRE 1993. "Aggregate level Simulation Protocol (ALSP) Program Status and History." MITRE Corporation, 7525 Colshire Drive, McLean, VA, 22102 (Mar.). URL: ms.ie.org/alsp/89-92_history/89-92_history.html

Nwana 1996. Nwana, Hyacinth S., "Software Agents: An Overview," In *Knowledge Engineering Review*, vol. 11, no. 3, Oct./Nov., 1996, 205-244.

Ourston 1994. Ourston, Dirk and Raymond J. Mooney. 1994. "Theory Refinement Combining Analytical and Empirical Methods," *Artificial Intelligence*, 66, 273-309.

Surdu 1998. Surdu, J.R. and U.W. Pooch. 1998. "A Methodology for Applying Simulation Technologies in the Mission Operational Environment." In *Proceedings of the 1998 IEEE Information Technology Conference* (Syracuse, NY, September 1-3). IEEE, Piscataway, NJ, 45-48.

Surdu 1999a. Surdu, J.R., G.D. Haines, and U.W. Pooch. 1999. "OpSim: A Purpose-Built Distributed Simulation for the Mission Operational Environment." *In Proceedings of the 1999 International Conference on Web-Based Modeling and Simulation* (San Francisco, CA, 17-20 January 1999). SCS, San Diego, CA, 69-74.

Surdu 1999b. Surdu, J.R. and U.W. Pooch. 1999. "Connecting the Operational Environment to Simulation." *In Proceedings of the 1999 Advanced Simulation Technology Conference* (San Diego, CA, 11-14 April 1999). SCS, San Diego, CA.

U.S. Army 1996. *U.S. Army, FM 100-6: Information Operations.* 1996. Headquarters, Department of the Army, Washington, DC