Connecting the Operational Environment to a Simulation

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. It focuses on the synchronization of the real operation with the planned operation in simulation. This methodology uses intelligent agents, called Operations Monitors, to compare the real operation to the plan, which is running in an operationally-focused simulation. When the Operations Monitors detect differences that threaten the success of the plan, they advise the decision-maker. This paper focuses on the problems of keeping the real operation and the simulation of that operation synchronized so that the Operations Monitors can accurately compare them.

INTRODUCTION

A large number of tasks performed by commanders and staffs can be facilitated during operations by 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.

In this paper we do not elaborate on the uses of simulation for COA development and analysis, nor do we discuss the use of simulations for rehearsals. (Surdu, Haines, and Pooch (Surdu 1999) discussed these uses of simulation.) Instead we assume that these processes have been completed and that the operation is about to commence. We discuss 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. We propose a methodology in which simulation technologies support commanders and staffs during actual operations. We discuss a number of the technical issues arising from this methodology. In particularly we focus on the problem of synchronizing the simulation with the operation to facilitate their comparison.

USES OF 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 could be a useful tool in each of these phases.

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

UNSUITABILITY OF TRAINING SIMULATIONS

The military community has developed a large number of simulations for training and analysis. 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 1999) enumerated the desirable capabilities for an operationally focused simulation to be used during operations, and 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 that paper Surdu, Haines, and Pooch described a simulation they built that meets these requirements; however, this methodology does not rely on that simulation. Any simulation that meets these requirements supports this methodology.

One government off-the-shelf simulation product that does not currently have all the properties described but might be appropriately modified to do so 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" to manage the receipt and answer of subscriptions and queries. ModSAF is not inherently cross-platform, but the GUI (which communicates with the simulation via UDP/IP messages) might be rewritten in a language like TCL/TK or Java to provide this capability.

PROPOSED METHODOLOGY

Our 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

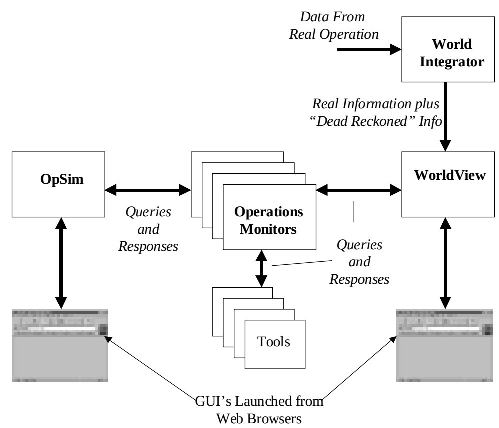

Figure 1: Proposed Methodology

directly for information.

Operations Monitors (OMs): The OMs are the heart of this methodology. They perform two important functions. They can take information from WorldView and update the state of entities in OpSim, seamlessly re-synchronizing the simulation to the real world. More importantly, they monitor the progress of the simulation and compare that progress with the progress of the real operation. When they discover significant deviations between the real world (WorldView) and the simulated world (OpSim), events that we call Potential Points of Departure (PPDs), they launch one of a number of 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.

Also, it is important to note that 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 and inform decision-makers in advance 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 becomes "low enough" the OM must inform the decision-maker.

OMs are implemented as intelligent agents. While there are a number of useful definitions of intelligent

[‡] 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.

agents, Nwana proposes that intelligent agents are processes that have autonomy, can collaborate with other agents, and are intelligent (Nwana 1996). We use this definition in the context of OMs within this methodology.

Each OM is interested in only a narrow domain. One logical taxonomy of OMs might be along the lines of 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. 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. For some OMs, the PPDs may reside in a fixed knowledge base (which may be in the form of rules) (Giarratano 1989), while the PPDs for others may be in the form of some refinable domain theory (Mitchell 1997). Another promising candidate is the fuzzy inference engine 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 make the job of the OMs easier, 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 system, 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, 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, the Terrain OM might notice that a bridge along a unit's planned avenue of approach has been destroyed. The Terrain OM might then call a path-planning tool to compute an alternate route. This path-planning tool can be simple or complex, possibly taking into account speed of travel, cover and concealment along the way, and/or other information. When the Terrain OM receives the answer from the path-planning tool, it decides whether this new route adversely impacts the timetable of the plan. If so, it sends a warning to the commander.

Similarly 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, the OM can call a combat attrition model to determine the difference in expected losses. If it appears that this difference will adversely affect the plan, the Enemy OM will notify the decision-maker.

If the Mission and Time OMs note that some 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 is conducted through a message-passing protocol. There are two kinds of requests: 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.

Some closing comments on this methodology: Note that normally OMs do not make tactical decisions. The purpose of an OM is to 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.

SYNCHRONIZING THE REAL OPERATION WITH THE SIMULATED OPERATION

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 commences, the plan will certainly diverge 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 rating 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 we allow the simulated plan to continue 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 where the operation will be at some future time. The problem, however, is to define a synchronization mechanism which is feasible and adaptive. The approach we propose is best illustrated through two examples.

The combat effectiveness of entities (units) in the simulation are characterized by some probability distribution(s) with appropriate parameters. These probability distributions (which might be different for different classes of entities) were used in the analysis of the various COAs during the planning phase, and they were also used to simulate the interactions between the various entities as the simulation was paralleling the operation in near real time.

At the time the OMs determine that the real operation and the planned operation are significantly different, they have a body of historical data on the actual effectiveness of the classes of entities. The OMs must do two things: update the current states (e.g., the number of casualties) and update the future performance of the entities within the simulation. An OM can try to refit the historical data to the family of probability distribution described for that class of entity. There a number of distribution-fitting packages on the market which accomplish such a task. They usually use maximum likelihood estimators (MLEs) to recommend some distribution for a set of data based on which distribution has the smallest p-value or sum of squared errors. When the OM tries to refit the historical data gained thus far in the real operation to the family of distribution defined for the combat effectiveness of the entity, some p-value or sum of square errors will be generated.

If the measure of error is below some threshold, the OM will conclude that the distribution family is correct, but that the parameters (e.g., μ and σ for a normal distribution) are incorrect. In this case, the OM would automatically send an update message to OpSim. As this should make the future performance of the simulation better over time, this updating scheme makes the system adaptive. If the error metric is above that threshold, the OM will conclude that the family of distribution chosen is incorrect. At this point, the OM will open a dialog with the user of the system describing the problem. It is then up to the user to decide whether to merely update the parameters of the current distribution with the best possible values, change the family of distribution used, or determine that the results are anomalous and decide that no update is necessary.

The preceding illustration shows adaptive updating of OpSim for combat effectiveness; however, the methodology is valid for other characteristics as well. Movement rates of classes of entities are also defined in terms of probability distributions (e.g., an average movement rate and some variance). If all the entities (units) of a certain type are moving more slowly than predicted, we would want to do two things: put the entities in their current locations in the real operation and adjust the parameters on the distribution which describes their movement rate. Again, it would be up to the decision-maker to determine if this difference was anomalous (e.g., unseasonable weather, entities beginning movement late, etc.) or required an update to the simulation.

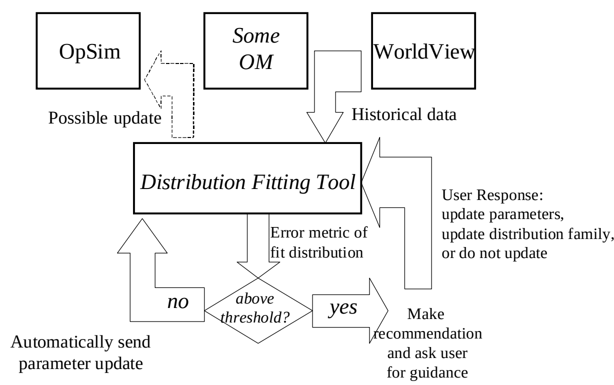

Figure 2: General Depiction of Synchronization/Updating Scheme

Another, less-technically interesting update of OpSim would occur when the number of entities was significantly different. If, for instance, the plan assumed that the enemy would have three tank battalions, but WorldView indicates the enemy actually has four, OpSim would need to include this additional unit in the future. Adding this unit automatically would be difficult, since an intelligence officer would have to provide OpSim with an estimated plan for this new entity. Initially, the OM might provide OpSim with a plan that extrapolates the entity's velocity vector, but adding new entities would probably need some human intervention.

The basic idea, therefore, is for one or more OMs to analyze the performance of the simulation with respect to the real operation. The OMs can make small updates in the parameters of OpSim automatically. For larger deviations they query the user for help. Just how much data needs to be gathered before the differences are significant enough to justify modifying OpSim is unclear. One problem with analysis of military operations is that the experiments are not reproducible, much of the area of operations is destroyed in the process, and many of the witnesses are killed. One approach to dealing with this issue is for the threshold (used to determine whether to update automatically) to be adaptive. The OMs can use the performance of the simulation after an update to help it adjust the threshold. At some point it might also be appropriate for the correctness of the human user's decision to be used to adjust this threshold as well.

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.

FUTURE WORK

A prototype OpSim that allows subscription to information by OMs has been built (Surdu 1999). 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 be HLA compliant with the ALSP Federation (MITRE 1993).

The ability for agents to query the simulation is of vital importance to the implementation of this proposed methodology. Consequently this capability is receiving special emphasis. A number of possible methods are being explored, including a derivative of KQML (Bradshwa 1997) as well as in-house methods. Once the simulation is completed, we will begin work on the various OMs.

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.

SUMMARY

We have proposed 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 representation 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.

In addition to advising the decision maker when the success of the plan is in jeopardy, another role of the OMs is to help the simulation adapt. They do this by analyzing the historical data of the current, real operation and trying to fit this historical data to the family of probability distribution defined for the various classes of entities in the simulation. If the OMs can fit the historical data to the defined distribution, they update the parameters of that distributions in the simulation automatically; otherwise, the OMs ask the user to update the parameters, change the family of distribution, or make no update at all.

This updating of the simulation is crucial to the success of the methodology we have 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.

REFERENCES

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

Bradshwa, J.M. 1997. "KaoS: Toward An Industry-Strength Open Agents Architecture." 1997. *Software Agents (DRAFT)*, J.M. Bradshwa, ed. AAAI Press.

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.

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.

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 1999. 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.

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