A Methodology for Applying Simulation Technologies in 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

Abstract-- This paper proposes a methodology for applying simulation technology to the monitoring and control of operations. This methodology includes the use of an operationally-focused, aggregate-level, discrete-event simulation. Within this methodology the progress of the real operation and the simulated (planned) operation are compared by intelligent agents. These agents can launch additional simulations and other tools to explore differences between the ongoing operation and the plan. Based on the results produced by these agent-launched tools, appropriate advisory actions are taken.

Keywords -- simulation, operation monitoring, intelligent agents, command and control

I. 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 focused 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.

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. 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 impending problems. Finally we propose a methodology in which simulation technologies support commanders and staffs during actual operations.

II. 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). This involves proposing a number of possible means of accomplishing the mission. At the same time, the staff identifies the most likely and most dangerous enemy COAs. With an operationally-focused simulation, the staff enters the enemy and friendly courses of action and simulates them. After several iterations, the simulation tool rates the various COAs using specified decision criteria. These ratings would be an additional source of information the commander might use in choosing which COA to pursue.

Once a COA has been chosen, it is developed into a full plan. The simulation might also facilitate this process. Once the plan is finalized, it can be played back for the commander, his staff, and the subordinate leaders. At critical moments the simulation can be halted and the conduct of the operation at that point can be discussed. The simulation can also be rolled back or fast forwarded to the previous or next critical event, respectively.

Once 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. When significant deviations from the plan occur, tools are launched which explore the impact of these deviations. Finally the commander is advised of significant changes.

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 [1].

III. 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 work stations.

The desirable capabilities for an operationally-focused simulation to be used during operations are enumerated in [2], and they include:

- The simulation must be runable from a single work station 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.

IV. 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.

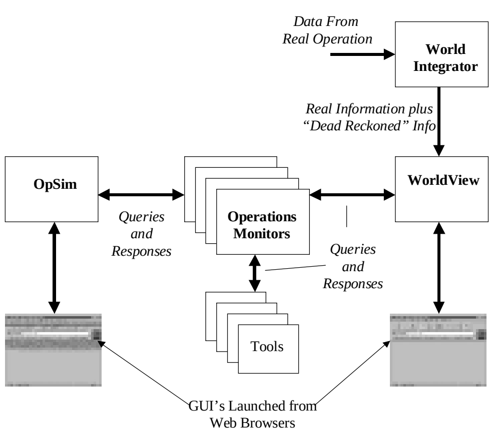

Figure 1: Proposed Methodology

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. They can take information from WorldView and update the state of entities in OpSim, seamlessly resynchronizing the simulation to the real world. More importantly, they monitor the progress of the simulation and compare that with the progress of the real operation. When they discover significant deviations between the real world (WorldView) and the simulated world (OpSim), events which 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. 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 when some time table 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 agents, the one we use in the context of OMs is that agents are software which have autonomy, can collaborate with other agents, and are intelligent [3].

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 [4, 5].

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 [6]. For some OMs, the PPDs may reside in a fixed knowledge base (which may be in the form of rules) [7], while the PPDs for others may be in the form of some refinable domain theory [4].

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.

Tools: We do not attempt to enumerate all possible, useful tools. Instead, we give examples of tools and how they might be used by the OMs. 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 time table of the plan. If so, it sends a warning to the commander.

Similarly, if the Enemy OM noted that during the planning, two squadrons of enemy aircraft were assumed to be in the area of operations. Based on information from WorldView that there are instead three wings (many more aircraft), it 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 commander.

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 API. There are two kinds of requests: individual queries and registration of interest. 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 that it wants to know if the strength of the units in a given area drop below some threshold. 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." [1]. 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 which occurs within the heads of the commander and his staff [8]. Our proposed methodology provides additional support for this process.

V. Technical Challenges

There are a number of significant technical challenges in integrating simulation into the running of operations. The principal challenges include:

An operationally-focused simulation, as described in [2], must be constructed.

There is a potentially large amount of state information that must be shared between OMs and the OpSim. Some of this information will need to be passed to tools that the OMs launch to explore PPDs. Shared memory approaches are generally not robust enough for tactical environments, and network bandwidth is often limited in operations centers.

As intelligent agents, the OMs must have some degree of autonomy. They must monitor the actual operation and the simulated operation and decide when to launch other tools to explore differences. This requires a great deal of intelligence and probably some mechanism for incremental learning.

The OMs must decide, based on the results of the various tools they employ, whether the new development significantly impacts the operation. This would probably use a scheme similar to that described for interface agents by Maes [9, 10].

The OMs must eventually collaborate with each other. While the initial prototype will concentrate on the piece-wise evaluation of the differences between the simulated and actual operation, it is likely that more useful information will be gained through collaboration between these agents. It is possible that one or two factors in isolation would not indicate danger but together might indicate a serious threat to the operation. It has not been determined whether these agents will collaborate in a peer-to-peer manner, through a facilitator, or in an OM hierarchy.

The OMs must eventually be robust enough to adapt their decision criteria to changing environments. While adjusting to natural environmental changes is important and not trivial, the more technically interesting adaptation involves other factors. As the enemy adapts to our tactics and we adapt to their, the effectiveness of certain plans may change. Incremental learning is still an open research issue, particularly in this domain.

The issue of WorldIntegrator and World View 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 a similar unit is reported by some sensor, 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.

VI. Future Work

We are currently building the operationally-focused simulation envisioned for this domain. It is an aggregate-level, discrete event simulation capable of near-real-time and faster operations. As an aggregate-level simulation, it will be designed to be compliant with a protocol like the DoD Aggregate Level Simulation Protocol (ALSP), and our intent is for it probably eventually to be HLA compliant with the ALSP Federation [11].

Ultimately users will connect to the simulation server via a Java applet from any web browser. The simulation will run on the server hardware, and the GUI will run via the browser. This allows users to launch the simulation on low-end hardware from locations that are remote from the simulation server.

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 [12] 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.

VII. 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.

VIII. References

[1] URL: mole.dc.isx.com/cpof and www-code44.nocs.mil/cpof.

[2] Surdu, John R., Gary Haines, and Udo Pooch, "A Purpose-Built Distributed Simulation for the Mission Operational Environment," Society for Computer Simulation, *1999 International Conference on Web-Based Modeling and Simulation*, San Francisco, CA, 17-20 January, 1998, vol. 31, no. 1.

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

[4] Mitchell, Tom M. *Machine Learning*. Boston: McGraw-Hill, 1997.

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

[6] Russel, Stuart and Peter Norvig, *Artificial Intelligence*. Upper Saddle River, Jew Jersey: Prentice Hall, 1995.

[7] Giarratano, Joseph and Gary Riley, *Expert Systems: Principles and Programming*. Boston: PWS-Kent Publishing Co., 1989.

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

[9] Maes, Pattie, "Agents That Reduce Work and Information Overload," Communications of the ACM, July 1994, vol. 37, no. 7, pp. 31-41.

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

[11] URL: ms.ie.org/alsp/89-92_history/89-92_history.html

[12] Bradshwa, J.M., ed. "KaoS: Toward An Industry-Strength Open Agents Architecture," *Software Agents*. AAAI Press, 1997.