Proceedings of the 2008 Winter Simulation Conference
S. J. Mason, R. R. Hill, L. Mönch, O. Rose, T. Jefferson, J. W. Fowler eds.

A MULTI THREADED AND RESOLUTION APPROACH TO SIMULATED FUTURES EVALUATION

David R. Pratt
Robert W. Franceschini
Robert B. Burch
Robert S. Alexander

Science Applications International Corporation
12901 Science Dr.
Orlando, Florida

ABSTRACT

As part of the DARPA Deep Green efforts, SAIC developed a multi-threaded and resolution approach to constructing and evaluating simulated futures to address the SimPath component. By making use of heuristically derived breakpoints in a provided plan, we can construct a series of possible futures that are grouped into what is referred to as the probabilistic fluent graph (PFG). Evaluation of the PFG is done through a multiple resolution and fidelity modeling system that applies computational resources to the areas of highest likelihood. In this paper, we will describe the architecture, processing, and data structures needed to bring this concept to fruition.

1 INTRODUCTION

At the most basic level, military operations can be divided into two phases: planning and execution. During the planning phase, the commander and staff develop a set of possible futures with the intelligence officer (commonly referred to as the S2) playing the role of the opposing force. This is largely a deliberate process that makes use of the commander and staff's experience and insights to develop a best course of action that is issued as a plan for execution in the next phase. The time constraint in developing this plan is largely driven by higher headquarters. Once the plan has started execution, the commander and staff must now recognize when the actual state of the world deviates from the plan and replan accordingly. Instead of having the luxury of time to think and play out the options in a deliberate manner, the commander and staff must react and develop the new course of action in a matter of minutes. As such, reliance on personal experience in analyzing the often incomplete data that is coming in from the field often is the only tool the commander and staff have for developing a new course of action and making decisions.

In military planning, the evaluation of potential futures is commonly referred to as the course of action (COA) analysis process. The generation and evaluation of these courses of action are some of the most time-consuming and subjective tasks in the mission planning process. Often done under severe time pressure, most staffs revert to their experiential base to select options and determine their viability. Since this is a subjective process, it is fraught with the typical issues associated with most human tasks. Recent advances in cognitive and computational sciences allows us to address some of the issues of the large number of options considered and conduct objective evaluations of them in a timely manner. This representational and processing element is the SimPath component of the DARPA Deep Green effort.

In the abstract, Deep Green is a commander and staff aid to help rapidly develop, evaluate, and interpret plans. It is critical to understand that it does not remove the commander or staff from the command and control process. Rather, as a cognitive and tracking aid, it allows them to more rapidly respond to dynamic and unfamiliar situations. It still relies on the commander and staff knowledge, experience, and insight to interpret, understand, and execute the recommended plans.

This paper will briefly discuss the overall Deep Green system and then focus in on the SimPath component. Specifically, we will cover the representation and execution of the possible future plans. While the issue of the environmental representation is a critical part of the overall SimPath effort and consistency of results, to do the topic any justice is beyond the scope of this paper.

2 DEEP GREEN ARCHITECTURE

At the highest levels, Deep Green is comprised of three interrelated components, shown in **Figure 1**. The commander and staff interact with the Commander's Associate while the Crystal Ball and SimPath elements operate in the

978-1-4244-2708-6/08/$25.00 ©2008 IEEE

background. The Environment referred to in the figure below is the Command and Control (C2) environment in which Deep Green operates. It provides the context in terms of inputs from (FBCB2), Command Post of the Future (CPoF), and Publish And Subscribe System (PASS). Likewise, orders are sent to subordinates via standardized interfaces to the C2 systems.

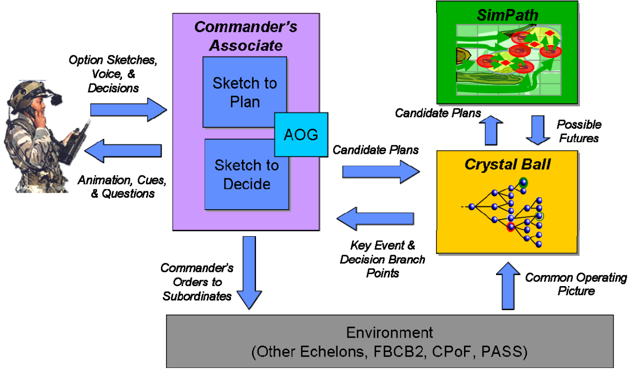

Figure 1: The Deep Green Architecture

Commander's Associate (CA) is the Warfighter Computer Interface component for Deep Green. The sketch-based interface uses more natural human interface interaction styles for capturing a Commander's intent. To do this effectively, the system must operate in the Commander's context for interactions and must make intelligent inferences about the intent and the execution details of the plan.

The functions of Crystal Ball are to build and maintain the Futures Graph, keep the Commander well-supplied with good options, and eliminate blind alleys. The SAIC team's innovative approach is to implement the Futures Graph as a probabilistic fluent graph (PFG), which provides a common Futures Graph for Commander's Associate and SimPath, as well. Since the generation and management aspect of the PFG and the simulation process do not have clear sections, there is not a clear conceptual boundary between the Crystal Ball and SimPath. As such, we will be discussing some elements of both Crystal Ball and SimPath in this paper in order to provide a comprehensive view of how we manage the simulated futures.

SimPath is the synthetic battlespace engine that produces correct and qualified predictions of future activities based on the Commander's expectations. It must provide accurate generation of topical and relevant optional actions based on current intelligence and COE experience. It must be able to support predictive models that unfold accurate futures within challenging timeframes. Accurate predictions must be coupled with learning and the ability to predict tactical creativity as seen in current conflicts. SimPath must be able to support all aspects of the COE and be able to predict at both the macro- and the micro-level, dependent on the level of the Commander's abstraction into the plan. As Deep Green is employed across different echelons, SimPath must be able to support various abstractions in a manner that correlates up and down the command hierarchy.

3 PLAN REPRESENTATION

For the sake of simplicity, we are going to assume a single-level plan representation. In reality, each of the plan nodes can be decomposed down to multiple subordinate plans. While this fractal nature is a key element of the C2 process, and a topic of follow-on work, we can discuss single-level plans without loss of generality. Likewise, other than synchronization points, we are not considering adjacent and supporting units plans. The follow-on hierarchical version of Deep Green will address them as well as part of the hierarchical planning process.

At the lowest level, a plan is constructed of a series of task nodes or fluents, shown in **Figure 2** (Cohen 2001). At the level we are planning, these represent atomic actions. The contents of the nodes can be divided up into roughly three categories. The static information contains the Task Name, Activity, Resources, Preconditions, estimated duration, and stopping condition. These are planned during the initial planning process and could be updated during the replan process. Of key interest is the Stopping condition that signifies a planned transition out of the current task. This could be time-driven (Attack at 1000), geographic-based (When reaching a phase line), coordination-driven (Link up with 2^{nd} squad), event-driven (When you take enemy fire), etc. Obviously, a time or distance transition will provide a better Estimated Duration value.

Task Name
Activity
Resources
Precondition
Stopping Condition
Est. Duration
Percent Compete
Next Task (s)
Likelihood of task (s)

Figure 2: Task Node / Fluent Representation

The dynamic portion of the plan is Percent Complete slot. As the node is being executed, the value is updated to project progress to the planned stopping condition. This value provides a metric into the status of the plan and allows the commander and staff to monitor the execution progress.

The remaining part of the plan node is of variable length. Initially, as shown in **Figure 3**, there are follow-on single tasks to each task. This is consistent with the military planning process, where that plan is a linear sequence of tasks. However, as SimPath executes, possible alternative futures are generated through plan perturbation (initially, this is largely a geographic consideration such as taking a left or right at a road junction) or interaction with the enemy (spotting, attack/defend, etc.). This results in a possibility of having more than one possible follow-on task. Thus, the link to the next task and the probability of taking that link are stored with the task node.

As shown in the figure above, there are three categories of task nodes. Previous nodes are nodes that have been executed and are preserved for historical and After Action Review (AAR) purposes. The Current node is highlighted and the one that the commander and staff believes we are currently in. This task is the one currently being executed and updated. This node serves as the root of the futures tree. The Planned future nodes, shown in blue are, as the name implies, the planned future tasks and have not been executed.

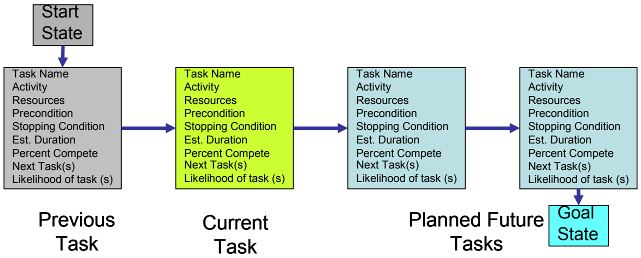

Figure 3: Plan Representation

The Start and Goal States are just tags to indicate where the plan begins and the expected outcome if the plan executes as anticipated. Since we will be discussing the operation of SimPath during actual events, for the remainder of the paper we can assume that we are in mid plan. This allows us to assume that the initial plan has been developed and the execution of it has started. Likewise, we have not reached the goal state, so successful termination is not an issue.

4 FUTURES REPRESENTATION

As discussed above, the initial plan is a sequential organization of the task. One of the old adages is “no plan survives first contact with the enemy.” Well, this is not quite true; many plans do not even make it that far. So experience has taught us that the future is much more like what is shown in **Figure 4**. The PFG can best be thought of as a directed cyclic graph rooted at the current node.

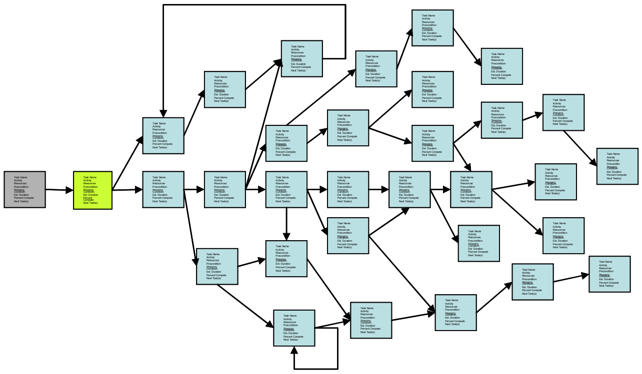

Figure 4: The Potential Futures Graph

There are several key things to note about the PFG. First is that it is cyclic. There are situations where a node loops back on itself (bottom center of the picture). An example would be “Conduct patrolling operations until enemy contact is made.” This is different from a holding task such as “establish a defensive posture,” in the sense that the unit subordinates are actively doing things that are repetitive. Larger loops, like the one shown in the center, are when a series of tasks link back on themselves. An example of this would be to clear out a building and then have to come back and do it again after the adversary had reoccupied it. These loop constructs are quite common for asymmetric and peace keeping operations where it seems that some activities repeat.

Also shown in the figure is the dead end or “blind alley.” As shown in the center top of the figure, these are one or more tasks that lead to a terminator that is not the goal state, and there are no more viable tasks that can be performed to reach the goal state. Obviously, these are options that, if known about in advance, should be avoided.

The final construct to note is the placement of the arrows. The majority of the tasks have the arrow leaving the right side and entering the left. This is to indicate normal commencement and completion of the task. As shown in a few of them, the arrows enter or leave the top or bottom. The leaving indicates a task that was interrupted prior to completion and another action commenced. An example of this is the enemy setting up an ambush along a trail. Likewise, when the task is entered through the top or bottom, it is joined in progress without preparation.

The two hardest issues to deal with are accurate predictions and timely execution. As the Commander’s plan progresses, variability and decisions occur that may permit alternative future paths. Accurate predictions of the variability and potential decisions are modeled. Each point of variability assumes particular conditions and can be assigned initial assessments based on analysis of world state against those conditions. These assessments are used to determine if there needs to be a parallel split to the world state.

As part of the construction of the PFG, we make use of Multi-Trajectory Control to help guide the generation of new trajectory and tasks. Multi-trajectory simulation futures heuristics are embedded in the branch-chooser capability. During simulation of a trajectory, a branch-chooser is invoked at the conclusion of a task or at the arrival of an event (subsumption) and will choose between several alternatives.

The first alternative is that branching may be "turned off" for that type of branch. In this case, the branch-chooser does not generate a node for the Futures Graph. It invokes a deterministic or stochastic response in the execution of the simulation, according to the features of the underlying simulation model.

The second alternative is that the branch-chooser may determine that a trajectory should be "pruned." In this case, the simulation will stop the execution of that trajectory and add a node to the Futures Graph, with information about the applicable state and an explanation of the termination criteria.

The third alternative that the branch-chooser would make would be to determine the level of fidelity of the multi-resolution model used to estimate future states emanating from the node in question.

From this discussion, it is easy to see that the definition of a task and the linking of the task are central to the execution and management of the PFG. To that end, we have developed a set of rules for the construction and linking of tasks:

- Tasks should be atomic for the level of unit being planned.
- No decision that could result in different follow-on tasks should be made in task.
- If a branching point is established within a task, the tasks is broken into two tasks.
- Linear tasks are not combined into a compound task.

From these rules, it is easy to see that even for simple military operations there will be a large number of tasks. We have made an overt trade-off in the number of tasks for simplicity of the task itself. This follows the agent paradigm:It is easier to manage a lot of simple well behaved things than it is to mange a few complex things with lots of exceptions. Thus, as SimPath executes, new states are added and old non-reachable states are pruned from the PFG.

5 THE AUGMENTED PFG

The PFG provides a good representation as to what might happen, but is actually quite sterile in its meaning. For that reason, we have augmented it with some additional plan-level information. As shown in **Figure 5**, the coloring of the futures boxes allows us to rapidly see which of the states fall into the three desirability categories. A simplistic taxonomy of Desirable, Borderline, and Undesirable provides the commander and staff an easy way of annotating each task as to how they feel about executing each of them. In some cases, there is a natural deterioration of the tasks. In others, things go bad quickly.

There is a series of green threads shown in the figure. These are the potential futures as represented by the linear plans. These are the trajectories the future might take. While we can not be certain any one of them will be correct, together the threads weave a tapestry that represents the likely future. Additionally, the trajectories prove useful as an organizing concept for the simulation excursions. In essence, these provide the scenarios for the simulated runs executed by SimPath. As each of these threads is executed multiple times, using standard design of experiments methodologies, the next probabilistic values for the next state can be derived. Likewise, if there is a trend to branch to an unplanned or unforeseen state, a new task can be added to the PFG.

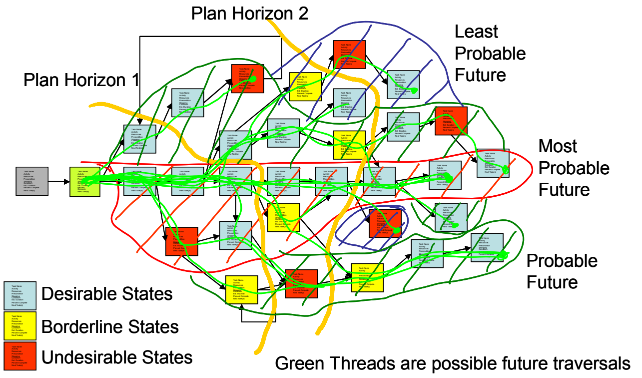

Figure 5: The Annotated PFG

Shown by the different color crosshatching, the likelihood of executing a task is also divided into three categories: Most Probable, Probable, and Least Probable. As we all know, there are some potential futures that are not very likely, but the consequences of that action are severe and need to be considered. The probability of the path can be derived using the Likelihood of Task fields for the traversed tasks.

The final annotations are the two Plan Horizons. These are how far out we are planning with a degree of certainty. They can be expressed temporally (the next 72 hours) or based upon some task attainment (when we get to Phase Line Delta). Either way, the further out we look, the less certain we are as to the correctness and viability of the plan. So, while we should consider these tasks, the amount of resources we should spend evaluating them is reduced. In our design, we have implemented two Plan Horizons since this matches with our evaluation process. Note that the Plan Horizon lines do not break up tasks, rather they assume the completion of the task. As the system matures, we expect to be able to push the plan horizons to the left and provide increasingly more valuable and insightful plan evaluations to the commander and staff in a timely manner.

6 FUTURES GENERATION

The Multi-Resolution Modeler subcomponent of SimPath accepts a set of plans and creates the qualified futures space. The core of the multi-resolution modeler is a multi-trajectory simulation implemented by incorporating multi-trajectory branching capabilities within the Department of the Army's OneSAF® (PEO STRI 2008). OneSAF, as the simulation engine for the entity and aggregate models, will have to be extended to support the task primitives and plan structures. The introduction of the branching and probabilistic considerations are two of the elements we will have to insert. We have not finalized the selection of the model for qualitative modeling portion, but might use the OneSAF simulation engine.

The rationale for the multi-resolution modeling is driven by the number of trajectories, the size of the PFG, and the number of iterations required to produce a statically valid response. It is constrained by resources and timing requirements. This leads us to the construct shown in **Figure 6**. At this time, the probability numbers shown for the futures are notional.

	Prior to PH1	Prior to PH2	Post to PH2
Most Probable Future (>50%)	Entity Modeling	Aggregate Modeling	Qualitative Modeling
Probable Future (50% > X > 25%)	Aggregate Modeling	Qualitative Modeling	Qualitative Modeling
Least Probable Future (<25%)	Qualitative Modeling	Qualitative Modeling	Qualitative Modeling

Figure 6: Futures Evaluation Schema

This schema allows us to allocate our computational resources in a manner where we have the greatest return on investment. What we are doing is putting the cycles where we have the most certainty of having the task take place. Since we have vague notions of what will happen the further out we look, it makes less sense to apply precision to analyzing these tasks. However, there has to be a third plan horizon that is the cutoff point for the simulation. The simple cutoff point is anything after the goal state or below a minimal probability of occurrence does not get simulated. As time elapses and the plan is executed, the probability of selected nodes will increase, and they can be expanded at that time.

6.1 Entity Modeling

The modeling of discrete entities in the battlespace is the traditional OneSAF space. In this realm, the physical constructs (tanks, soldiers, airplanes, etc.) and their interactions are explicitly modeled. Intuitively, it is the most realistic and understandable of the simulation classes. It is also, by far, the most computationally complex.

6.2 Aggregate Modeling

Aggregate modeling takes a slightly different approach in that it models the logical entities in the battlespace. Most commonly, these are the units (squads, platoons, companies, flights, etc.) that are involved in the action. We implemented aggregate models in OneSAF making use of the entity composer and developing simplistic Lanchester equation based attrition for the models. (Pratt 2006)We made use of the OneSAF terrain without modification. We are using that same approach for Deep Green.

6.3 Qualitative Modeling

One of the best analogies for qualitative modeling is the Risk board game (Wikipedia 2008). It is a fairly simplistic model that uses a basic terrain representation and arbitrates the interactions via a comparison of the relative strengths and a roll of five dice. While our version will be slightly more complex than the game, it will be significantly simpler that the other two classes of modeling. Using simplified pixel-based terrain for movement and intervisibility and Lanchester-based interactions, the resulting modeling, when executed over multiple times, proves a reasonable, low fidelity approximation of the possible futures.

These models have many advantages; however, in some situations, their predictive power is limited by various modeling effects and human ingenuity. For these challenging situations, we use a Motivational and Unconventional Local Effects (MULE) module based on the predictive technology underlying the Real-time Adversarial Intelligence & Decision (RAID) adversarial reasoning module (Kott 2005). MULE produces local (i.e., low-echelon) combat simulations that incorporate human motivational factors (e.g., tendency to fight/run), unconventional tactics (e.g., hiding among civilians), and non-linear local effects (e.g., effect of specific local terrain) based on observed historic patterns.

MULE is applied to classes of states the PFG probability weighting indicates likelihood are relatively poor. For these situations, MULE executes a local situation assessment where local combat factors, such as entity distribution, entity activity, and high-value local terrain, are disaggregated from the higher-level model using historic patterns of local behavior. These local situation factors are used as context for two predictive processes: motivation assessment and a local effects simulation, both of which expand on previous intent recognition and combat simulation work done in RAID and other programs. While not perfect, the inclusion of the "soft factors" helps balance the traditional bias of qualitative modeling to physical attributes.

6.4 Simulation Cloning

To evaluate the PFG in the time allotted, the sequential execution of the models using brute force is not viable, even using the multiple resolution modeling techniques described above. We have to introduce a degree of parallelism, running of the models is not supported in traditional simulation execution. For this, we turn to simulation cloning (Hybinette 1997). Simulation cloning enables the creation of several new copies of a running simulation sharing a common starting point and each modeling a different trajectory. The clones are inserted into the ongoing simulation as branch points, and each new clone proceeds in parallel. Here, a branch point defines the point in a simulation thread at the completion or interruption of a task when the simulation model is cloned; thus, the original and cloned simulation is then free to execute along different execution paths.

Simulation cloning improves simulation performance in two ways. First, simulations that are interactively cloned at a branch point are able to share the same execution path before the branch point, so the computations before the branch point need to be executed only once. Second, after cloning, multiple versions of the simulation may be able to share a single execution of certain computations that are common among them. Additionally, clones can be pruned, i.e., deleted, when it is apparent that the COA they are evaluating will not be competitive with respect to others. The cloning framework will include three components: 1) a situation database built from data feeds that characterize the current state of the system under investigation; 2) a faster-than-real-time forecasting simulator; and 3) an interactive environment that interfaces with both the database and the forecasting tool for what-if evaluation of alternative COAs.

6.5 Parallel Processing Model

The combined use of multi-resolution modeling and simulation cloning allows us to make use of parallel processes to execute the threads concurrently. In the early phases of the PFG evaluation process, the entity-based simulation model dominates the processing resources. Based upon our experience, these models run quite effectively distributed in a load-balanced, conservative time management scheme. As the model reaches the Plan Horizon 1 or the less likely nodes, the simulation is cloned and the lower single-threaded, lower resolution models are used. Since these models are single threaded, we can run them in discrete event as fast as possible (AFAP) mode. Thus, we can run more threads and the same number of processors in a given time. This helps to account for the combinatorial explosion of tasks as the PFG grows.

In the parallel processing mode, we make use of the branch and rendezvous construct to provide a consistent evaluation of a given state before the results are cloned for further simulation.

7 CONCLUSIONS

Presented in this paper were some of our concepts and approaches used for the SimPath element of Deep Green, as of the writing of the paper. As an ongoing program, the concepts will evolve over time. But the basic approach is sound and elements of it have been proven on several different efforts. It is important to note that this system will not replace the commander and staff. Rather, it will augment their abilities and allow them to be more responsive and effective.

REFERENCES

Cohen, P. R. 2001 *Fluent Learning: Elucidating the Structure of Episodes*, Proceedings of the Fourth Symposium on Intelligent Data Analysis, Lecture Notes in Computer Science. Springer.

Hybinette, M and R. Fujimoto. 1997. Cloning: A Novel Method for Interactive Parallel Simulation. In *Proceedings of the 1997 Winter Simulation Conference*, ed. S. Andradóttir, K. J. Healy, D. H. Withers, and B. L. Nelson, 444-451. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Kott, A. 2005. *RAID Real-time Adversarial Intelligence & Decision making*, Briefing, Available at http://dtsn.darpa.mil/IXO/downloads%5C20050531RYO77287.ppt [accessed July 23, 2008]

Pratt, D.R., E. Towers, D.R. Shires, and K.T. Kick. 2006. Progressive, Multi-Resolution Course of Action Analysis. In *Proceedings of the 2006 Interservice/Industry Training, Simulation and Education Conference*.

PEO STRI, OneSAF Home Page, http://www.onesaf.net

Wikipedia entry on Risk (game). Available at http://en.wikipedia.org/wiki/Risk_(game) [accessed July 23, 2008]

AUTHOR BIOGRAPHIES

DAVID R. PRATT (prattda@saic.com) is currently the Chief Scientist (Fellow) for SAIC's Analysis, Simulation, System Engineering, and Training Business unit. He also serves as the Forces Modeling and Simulation point of contact for DoD's High Performance Computing Modernization Program (HPCMP). A former Captain in the United States Marine Corps and Associate Professor of Computer Science, Dr. Pratt was the Technical Director for the Department of Defense's Joint Simulation System (JSIMS) program. He received a Bachelor of Science in Electrical Engineering from Duke University and a Master of Science degree and a Ph.D. in Computer Science from the Naval Postgraduate School.

ROBERT W. FRANCESCHINI (franceschinr@saic.com) is a Principal Investigator with Science Applications International Corporation. He earned a Ph.D. in Computer Science from the University of Central Florida. He has directed research leading to 50 scientific publications concentrated in the areas of computer generated forces, distributed simulation, and multi-resolution simulation.

ROBERT B. BURCH (burchr@saic.com) has over twenty years total work experience in Training and Simulation. His experience includes computer generated forces, crew cabin simulations, aircraft systems modeling, user interfaces, simulation services, and behavioral modeling. He has spent over fifteen years in a lead capacity generating technical approaches, design guidance, technical oversight, and planning. Mr. Burch was the Chief Scientist for the Close Combat Tactics Trainer Semi-Automated Forces and System/Software Architect for the Objective OneSAF System. Mr. Burch is currently the Chief Engineer for the Integrated Mission & Simulation Systems Operation. Mr. Burch has a BS in Computer Science from the University of Central Florida.

ROBERT S. ALEXANDER (alexanderro@saic.com) is a defense analyst working for SAIC. He retired from the Army in 1997, and has graduate degrees in Operations Analysis and Computer Science from the Naval Postgraduate School. The focus of Rob's work has been in aggregate-level simulations of combat operations. Among other projects at SAIC he has worked for DARPA on military Course-of-Action Analysis, for Natick Soldier Center as the lead analyst for the Future Force Warrior Advanced Technology Demonstration, and for National Cancer Institute on a project to develop distributed tools for cancer research. Dr. Alexander is also an adjunct professor at George Mason University where he teaches a course in Models, Gaming and Simulation.