Progressive, Multi-Resolution Course of Action Analysis

David R. Pratt, PhD
Ellen Towers
Science Applications International Corporation
Orlando, FL
[Prattda | Ellen.B.Towers]@saic.com

Dale R. Shires
Kelly T. Kirk
Army Research Laboratory (ARL)
Aberdeen Proving Ground, MD
[Dale.Shires | Kelly.T.Kirk]@us.army.mil

ABSTRACT

We are in the process of developing a Joint, progressive, multi-resolution course of action analysis (COAA) capability. This capability augments government off the shelf (GOTS) simulation tools by using a well-defined course of action (COA) data interface in keeping with the Department of Defense (DoD) vision for GOTS simulation tools. In addition, it makes use of common synthetic natural environment (SNE) representations and reasoning algorithms through a well-defined data interface, consistent with the DoD vision for a common SNE. The progressive COAA tool allows planners to quickly identify candidate high-level plans, and then focus time and computational power on detailed exploration of the viable alternatives. The progressive COAA tool is broadly applicable for training, rehearsal, and real world operations. The approach provides a mechanism for standards-based data exchange between the COAA tool and GOTS simulation. By exploiting multi-resolution modeling (MRM) capabilities of the GOTS simulation, we provide a deeper understanding of the data interface needs for COAA and other components that need to link to the COTS simulation. In addition, the project will contribute further understanding to the needs of common SNE for MRM-based COAA. This effort expands on our internal research and development (IRAD) project on variable model fidelity for progressive course of action analysis of the Joint battlespace. We leverage our experience with a data-centric simulation architecture, MRM, and common synthetic environments, and our ongoing efforts supporting DoD and adapting our COAA system to the Joint context.

An early version of this paper was presented at the 2006 Spring Simulation Interoperability Workshop. This paper has been expanded based on comments and ongoing work.

ABOUT THE AUTHORS

DAVID R. PRATT is currently the Chief Scientist (Fellow) for SAIC's Strategies Simulation and Training business unit. As a vice president for technology, his responsibilities include developing and fostering leading- edge information technology and M&S technologies. He also serves as the Forces Modeling and Simulation point of contact for DoD's High Performance Computing Modernization Program (HPCMP). He received a Master of Science degree and a Ph.D. in Computer Science from the Naval Postgraduate School and a Bachelor of Science in Electrical Engineering from Duke University.

ELLEN TOWERS is a software engineer in SAIC's Training and Simulation Solutions business unit. She has a Master's degree in Computer Information Science from the University of Pennsylvania and a Bachelor's degree in Computer Science from the University of Central Florida.

DALE R. SHIRES is a team leader in the Computational Science and Engineering Branch of the High Performance Computing Division at the Army Research Laboratory. He also serves as a technical representative for the User Productivity Enhancement and Technology Transfer component of the DoD's High Performance Computing Modernization Program. He has over 10 years experience in parallel computing and computational science. He received both his Master's and Bachelor's degrees in Computer Science from the University of Delaware.

KELLY KIRK is a member in the Computational Science and Engineering Branch of the High Performance Computing Division at the Army Research Laboratory (ARL). As a member of the Sci Vis team, his responsibilities include developing and incorporating new technologies to provide enhanced ARL's visualization capabilities. He has over 10 years experience in 3D visualization and distributed programming. He received a Master of Science degree in Computer Science from Johns Hopkins University and a Bachelor of Science degree in Computer Engineering Technology from Spring Garden College.

Progressive, Multi-Resolution Course of Action Analysis

David R. Pratt, PhD
Ellen Towers
Science Applications International Corporation
Orlando, FL
[Prattda | Ellen.B.Towers]@saic.com

Dale R. Shires
Kelly Kirk
Army Research Laboratory (ARL)
Aberdeen Proving Ground, MD
[Dale.Shires | Kelly.T.Kirk]@us.army.mil

Introduction

As part of a series of ongoing collaborative research efforts conducted by SAIC and ARL into next generation simulation and high performance computing (HPC) applications, we are engaged in a process of applying simulation tools to the automation of the course of action analysis (COAA) process (Pratt 2006). This effort centers on making the process more efficient by presenting the planner with the evaluated results and recommendations for a series of auto-generated plan templates. Following the philosophy that each level of command is concerned with the command and control of units one echelon down; we used a series of plan decomposition techniques to disaggregate the units into a series of lower level units and eventually to the platform level. By its very nature the proof of principle described in this paper simplifies the problem of joint combat, and we believe the architecture is robust enough for further study. Additionally, we will discuss implementation of aggregate units as primitives into the OneSAF objective system as one of the key enablers of this approach.

Motivation

While the research is driven by the time and effort required to develop operational battle plans in the real world, we viewed that addressing that issue was a bridge too far for any small effort. Rather we determined that looking at the problem and solution architecture and how simulation could aid the planner in the selection of plan alternatives would be an appropriate research goal. In doing this, we realized that the "low hanging fruit" was the time and effort it takes to set up a simulation exercise. In using simulation to support COAA, this often becomes the driving cost and time element. Thus, it became our objective to produce a realistic set of executable plans given an operational scenario. While the plans might not be perfect, they should be good enough to provide the context for the adjacent and opposing forces and allow the event planner to augment their units with more detail.

The longer term motivation is to develop a COAA tool that will allow the planner to enter (or better yet have the data drawn directly from the operations command, control, communications, computer, and intelligence (C4I) system) the current operational state (to include the enemy position), and the operational goal; and from these initial conditions, develop a set of ranked operational plans that can form the basis of the next set of orders. Through the use of simulation and high performance computing (HPC) resources, the ability to perform rapid iterations on the plans and explore non-traditional plans would be greatly enhanced. Thus, both the timeliness and confidence in the orders will increase. While this a currently beyond the state of practice, programs like the Army's Future Combat System (FCS) that make use of embedded simulation connected via Network Centric Warfare systems to the next generation High Productivity Computer Systems (HPCS) are creating the infrastructure for such a system in the not too distant future.

Previous Work

The concept of automating COAA is not unique; one of the more recent examples is that of Project Albert (USMC 2006). This system primarily focused on the use of agent technology to evolve and understand complex behaviors and responses. As an agent-based system, planning and coordination was largely done at the entity level and proved it was possible to use a few simple behaviors, run over multiple iterations, to produce realistic looking scenarios. However, these were emergent vice top down doctrinal behaviors. For the purpose of this research, we are looking at going from the top down.

In (Tolk 2003) discusses the need for and ability of decision support systems (DSS) to support military planners. It is most interesting that Tolk does not take the traditional computer scientist view of the problem. Rather, he looks at it from the needs of the user. In (Tolk 2005) he describes how agent-based systems can support the interoperability requirements needed by such systems. In both cases, the discussion centers on the needs and the function of the DSS system to support COAA.

In his seminal paper, "The Base of Sand Problem," Davis discusses the use of simulation and some of the inherent limitations (Davis 1991). Sadly, while the paper is al-

most 15 years old, it is still as relevant to current state of COAA simulations as when it was written. As the paper points out, the human element in the evaluation and execution of the models is key. For this reason, we believe that any system designed for COAA should provide the planner recommendations and cannot be considered "the final answer" for planning process automation. In many ways, this is similar to the retrieval phase in Case Based Reasoning (CBR) Systems. A National Academy report, (Pew 1998), drew the same similarities between CBR and the planning process.

Likewise, this project is not the first to suggest the use of multiple resolution models (MRMs). In his tutorial, Davis presents some of the history and implications of MRMs (Davis 2005). It is interesting to note the reference to linking different models together. Since many of the models were developed in isolation from each other, there are key implementation differences that yield differing results at each level. Thus, while the models may feed each other, the differing algorithms used in the models and the creation / deletion of information calls the validity of the linkage into question. For this reason, we chose to extend the U.S. Army's OneSAF Objective System (OOS) and create aggregate models within the same framework as the entity instance. This effectively eliminates the simulation induced variations in the terrain, environment, and behaviors between the various echelons.

The elements of the proposed systems have their basis in well-referenced literature. However, as stated earlier, we did not find any references to the same model being used at different levels of aggregation or that the plans were linked at different levels to develop the recommended COAA.

Hypothesis

The basic hypothesis is that through progressive refinement of high-level auto-generated plans, a realistic plan can be developed. At each of the levels, three candidate plans will be developed based upon the best performing higher level plan. At the highest level, the plan consists of the starting location and the objective. At the lowest level, the plan consists of the defined routes for each of the entities in the battle space. The plans vary by the routes generated for the entities by each of the three route-planning algorithms. The resulting control flow is shown in Figure 1.

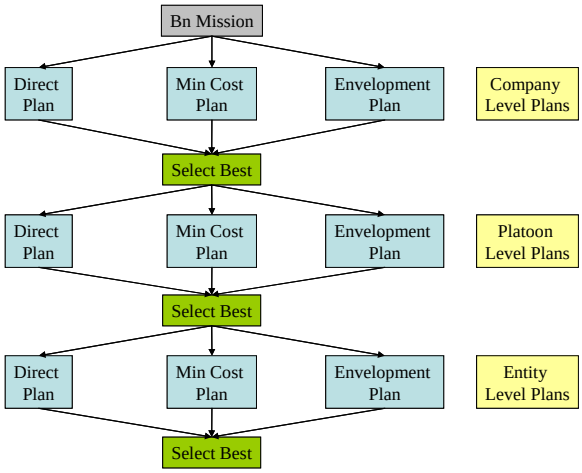

Figure 1. Decomposition Approach

The evaluation process is very similar to the selection and propagation methodology used in Genetic Algorithms. At each echelon and for each plan, multiple iterations are run using different random number seeds. The results from each of the runs are then computed and the like runs are combined to produce an average score. Combined with the natural non-determinism in OOS, this approach provides the datasets we will use for the data reduction, evaluation, and selection phase. At this point, the "best" plan is brought forward and decomposed.

In essence, we are implementing a directed search though the scenario solution space. The use of the evaluation function provides a pruning mechanism allowing us to evaluate nine cases (three at each level) rather than the full tree of 39 cases (three at Company, nine at Platoon, and 27 at Entity). By limiting the cases to three at each decision point, we have artificially constrained the number of possible alternatives. In an operational system, there would be considerably more choices producing a combinatorial explosion of possible cases requiring pruning of the case tree to reduce the problem to a tractable solution space. However, given that this a proof of principle, we felt that the constraint does not unduly limit the validity of the approach.

System Description

A common approach to analyzing candidate COAs is to execute them in an entity based training and analysis simulation system such as Joint SAF (JSAF) or the OneSAF Testbed Baseline (OTB). This technique requires a separate set up for each of the COAs. A largely manual activity, this process often results in a high time and computational cost of running multiple scenarios within these high-fidelity systems. One way to mitigate this cost is to employ variable levels of aggregation to simplify the number of simulated elements that must be

planned and controlled. As Figure 2 indicates, this technique tailors the COAA to use low-fidelity aggregate models quickly, using closed form simulation techniques in the beginning to trim the search tree. This series of runs executes faster than real time without a human in the loop (HITL). As the search tree is reduced, higher levels of fidelity and resolution are employed to obtain more accurate results. The final step in the process is to run real-time HITL executions to support final COAA, in essence, a mission rehearsal event. Prior to this MRM approach, different, and not always interoperable, tools are used as computer-based aids in the COAA process. Our approach makes use of a single simulation system that is capable of altering its fidelity and resolution to apply appropriate resources to the COAA problem in an efficient and timely manner.

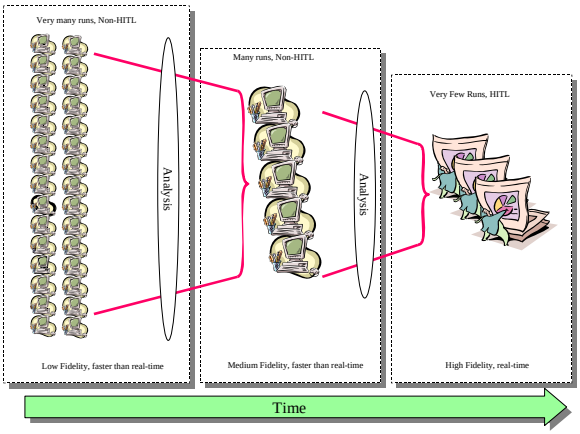

Figure 2 . Variable Fidelity over Multiple Phases

The use of multiple resolutions and fidelities in a composable simulation system such as the OneSAF Objective System (OOS) provides an important discriminator for this effort. Nowhere in our literature search did we find an approach that uses successive fidelity and aggregation increments to partition the solution using a single simulation platform. The OOS architecture and models are designed to support the composition of fidelity and resolutions ranging from simplistic, low-fidelity models to more complex, high-fidelity models of individual soldiers, vehicles, and building interiors. Composition of these capabilities is obtained either through application building or through parametric composition. In application building, separate executable versions of the tool incorporating different levels of functionality are included in a single application. In parametric composition, variable levels of fidelity are included in the models and activated by setting key parameters in the system at exercise run time. We take the viewpoint that the simulation system is independent of the COA tool, so we do not mandate a particular simulation system (such as OOS) but rather we define data interfaces between the COA tool and the simulation tool. This means that the COA tool could be implemented with any COTS simulation tool selected by the user, provided that the simulation tool supports the composition ideas outlined above.

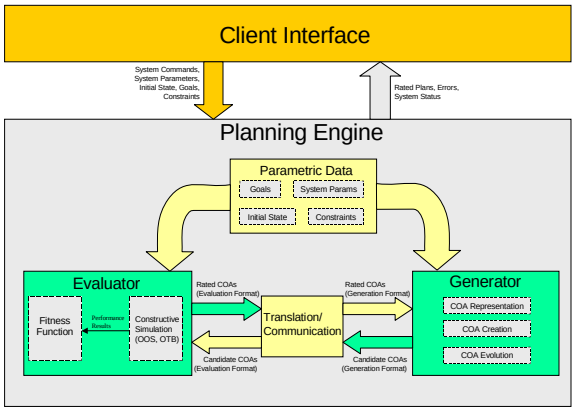

Figure 3. COAA Framework Conceptual View

Figure 3 illustrates the course of action analysis processes architecture framework as it stands today. We believe this fundamental architecture is sound for the larger operational context; the main extension needed is the Parametric Data for the Evaluator component (with the increased complexity of joint operations the evaluation of events is more complex and might involve second and third order effects). However, all components of the architecture will be extended to support Joint and non-kinetic operations. We have adapted the Planner and Evaluator to satisfy a simple Joint course of action requirements and develop support for synthetic environment analysis tools provided by a common synthetic environment representation.

Figure 4 presents a conceptual view of a complete system, which formed the road map for the IRAD prototype and the later joint ARL / SAIC effort. As indicated, there are several components to the system. A client interface allows the user to set parameters such as the scenario to be used, the number of plans to be created, and the duration of the planning session. The Planner and Evaluator (implemented in the IRAD prototype using OTB and extended in this project to encompass Joint operations in the OOS context) provide complementary functionality. Planning begins when the Planner creates an initial set of COAs, which it passes to the Evaluator. The Evaluator then executes the COAs, assigns a fitness, or "goodness," rating to each COA and returns the result to the Planner. The Planner decomposes the most promising COAs into lower echelons (higher fidelities and resolution). This set of COAs is passed back to the Evaluator and the cycle continues until some predefined termination condition is met. At the current time, the

stopping condition is the entity level. By working together in this way, the Planner and Evaluator allow creation and incremental improvement of candidate COAs: an evolutionary process intended to yield high-quality plans at a reasonable computational expense.

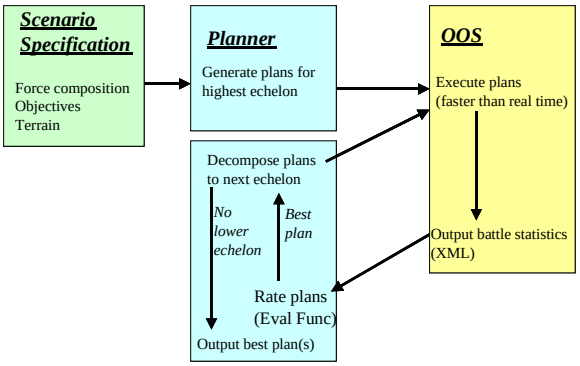

Figure 4. COAA System Conceptual View

Plans are made and evaluated starting at the highest echelon and entities are aggregated at each echelon level to reduce computation. For example, at the Brigade level, entities are aggregated as Battalions, and each Battalion treated as indivisible. Each aggregate entity has attributes such as strength or attrition. Plans are then formulated using a small number of these Battalion entities, with rules governing things like logistics and avenues of approach that pertain to a force of Battalion size. As described above, aggregating the planning at this low level of fidelity allows a larger space of possible plans to be evaluated in the Planner. The search space is pruned to include only the best of the resulting plans, and the planning moves on to the next lower echelon.

Components of the COAA Architecture

Major requirements of the COAA architecture include the ability to:

1. Identify a situational awareness package definition that can be sent to the Planner
2. Define a simple mechanism for the Planner to generate COAs
3. Identify a format for the COAs to be sent from the Planner to the Evaluator
4. Develop a way to integrate the Planner and Executor, including a communications protocol and implementation and testing of a communications mechanism
5. Provide a consistent simulation environment that reduces the simulation induced discrepancies between the echelons
6. Implement a consistent methodology for rating plans based on execution outcome

Functional Scenario

From its inception, this effort was targeted as a proof-of-principle of the process and architectural structures rather than an operational system. As such, we were able to bound the problem space to a joint close air support mission. This scenario includes a blue tank battalion with a section of supporting A-10s advancing to an objective point with the possibility of encountering a red tank company enemy force along the way. Figure 5 shows the notional lay down of the scenario.

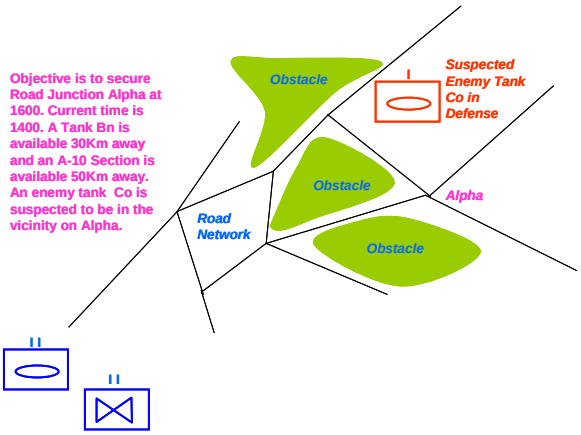

Figure 5. Notional Scenario

From some of our preliminary runs, the overall scenario using the direct path planning algorithm requires approximately 30 minutes of wall clock time and the red and blue forces do engage each other. As will occur in all OOS based simulation runs, there are some variations due to the simulation-induced variability. These are overcome by running each COA multiple times and averaging out the figures of merit.

Unit Levels and Models

The forces in Table 1 are part of the scenario described above. The rows indicate the various levels of decomposition we will be using for this project. The first row represents the traditional aggregate model where all forces are represented as units. Similarly, the third row represents the typical entity model in that all the platforms are explicitly presented. The middle row represents the most interesting case in that the entity aircraft engage the aggregate enemy tank platoon. The relative sizes of the aggregate units and entity-level simulated objects are shown in Table 2. The size of the aggregate units is based on the notional geographic extents of the unit when they are in a diamond formation. We treated the units as “blobs” and moved them as a single entity using OOS movement primitives.

Table 1. Simulated Battlespace Objects

Command Level	Ground Entity	Ground Entity Count[1]	Air Entity	Air Entity Count[2]
Battalion	Company	4/1	Section	**1**
Company	Platoon	16/4	A-10 Aircraft	**2**
Platoon	M-1 / T-80Tank	64/16 (blue/red)	A-10 Aircraft	2

Table 2. Sizes of the Simulated Entities

Unit	Width	Length
Company	614.24m	629.36m
Platoon	207.12m	214.68m
Entity	3.56m	7.34m

Route Planning

As discussed above, the three path planning algorithms are one of the two methods used to introduce COA variability. This section briefly discusses the three path planning algorithms. The other means of injecting variability is though the use of different random number seeds. This results in different random number draws generating different execution and results. We accounted for this variability by running each COA 10 times. Since the system is run in the single threaded closed form mode, the variability induced by human interaction or network delays is eliminated.

Direct Plan

The simplest of the three paths is the direct path. In this case, the entities proceed in a direct line from the origin to destination. As shown in Figure 6, there is a real chance this path will encounter some obstacles and result in the detailed reactive behavior of the entity slowing down and going around a restricted mobility area. Since the detailed path planning is non-deterministic, these lower level functions introduce the stochastic variability in the system. A sample dynamically replanned route is shown in purple in the figure.

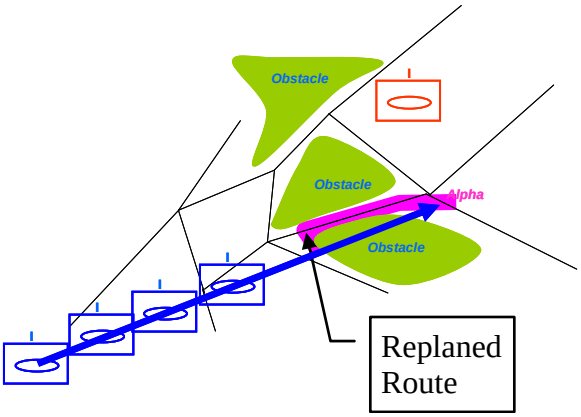

Figure 6. Notional Direct Path

Minimum Cost / Preplanned Route

As the name implies, this algorithm attempts to take the fastest route from the origin to the destination point using a minimum cost trafficability function. OOS uses a modified version of the A* search to find the route with the highest trafficability values; generating the shortest route in terms of time. As shown in Figure 7, the entities follow, in line, a direct path to the road and then follow the road and avoid entering the area marked "Obstacle." The contact avoidance and station keeping algorithms will introduce some variations, but are minimal. As shown in Figure 7, the avoidance of the obstacle is pre-computed for the assigned COA.

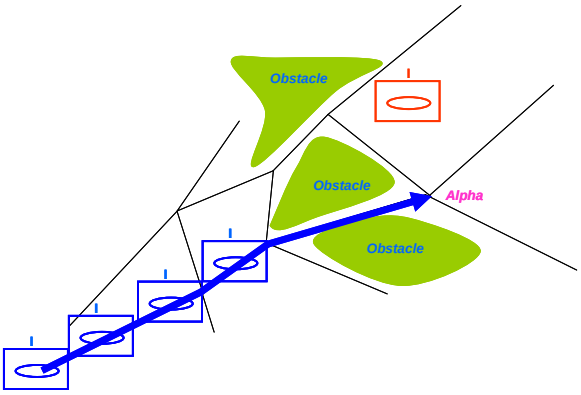

Figure 7. Notional Minimum Cost Path

While this approach takes longer to preplan, the thought is that these routes will exhibit less variability during run time and the time from crossing the line of departure to the objective will be reduced.

Double Envelopment

The final and most complex path algorithm is the double envelopment path. As shown in Figure 8, the force is split in half and uses a pincer movement to reach the

[1] The counts are listed in the form of number of blue entities / number of red entities at that given level.
[2] There are not any enemy aircraft in the scenario.

objective from both sides. The algorithm generates the turn points along the route as intermediate points based on the distance between the point of departure and the objective while OOS does the detailed planning and obstacle avoidance. While the paths are roughly equal length, the time to traverse them depends on the type of terrain and obstacles encountered. This could lead to the two sub units arriving at the objective at different times. For the sake of the evaluation function, we use the arrival time of the last unit to arrive.

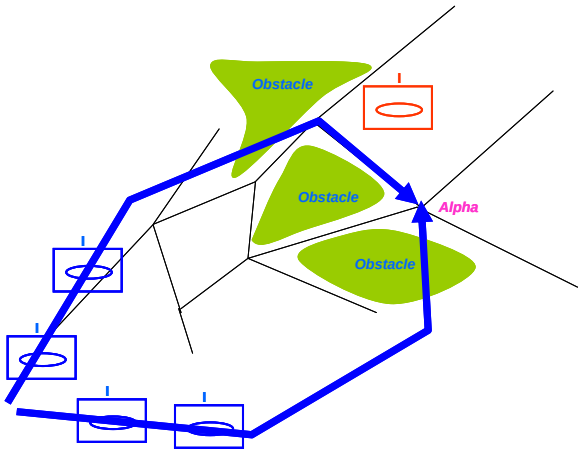

Figure 8. Notional Envelopment Path

Lanchester Equations

There are many arguments against the use of Lanchester equations due to their lack of representation of current operational reality. While many of these arguments are valid, these equations do provide a degree of simplicity for aggregate level interactions and have been used extensively to model force-on-force combat[3]. Thus, we chose to use the Lanchester equations, described in (Sidran 2006) and shown below, as the basis for the interactions between aggregate units with out loss of generality. We ran one turn per interaction.

$$\frac{dR}{dt} = -k_B(t), \quad R(0) = R_0$$
$$\frac{dB}{dt} = -k_R(t), \quad B(0) = B_0$$

R(t) and B(t) represent the strength at time t of the red and blue forces, and k_R and k_B the effectiveness of a red and blue individual, respectively. In this project we based strength on the number of weapons remaining in the aggregate entity. While each of the weapons represented one platform, we captured this as a parameter of the aggregate unit. Thus, at the company level, each entity had sixteen weapons and at the platoon, each entity had four. As attrition took place, the entity / weapon count was decremented to reflect the loss of a platform. When the entity had no more weapons, it was considered dead and no longer moved. We turned off collision detection to avoid the problem with the dead aggregate entities blocking each other.

Evaluation Function

As the scenario unfolds, time, space, and position information (TSPI) is logged to a XML formatted file, as are the interactions (weapon firing and impacts) and their results. This information is then post-processed and reduced. The resulting data is collated and used as the input for the evaluation function described below. We have set up the equation so that the smaller the figure of merit, the better the plan.

The generalized evaluation function is defined by the following scenario-specific parameters:

M = Mission Attainment (Boolean value if the mission was successful (0 = True, 1 = False))
T = Time on Target (the value is the delta from the estimated time to complete the operation normalized by the estimated time)
C_{kf} = Casualties count, killed, friendly
C_{ke} = Casualties count, killed, enemy
C_{ko} = Casualties count, killed, other
C_{wf} = Casualties count, wounded / disabled, friendly
C_{we} = Casualties count, wounded / disabled, enemy
C_{wo} = Casualties count, wounded / disabled, other
$k_{1..8}$ = Weighting constants for each of the terms (defined by the mission parameters)

The resulting equation is shown below.

$F(M, T, C_{kf}, C_{ke}, C_{ko}, C_{wf}, C_{we}, C_{wo}) =$

$$k_1M+k_2T+k_3C_{kf}+k_4C_{ke}+k_5C_{ko}+k_6C_{wf}+k_7C_{we}+k_8C_{wo}$$

The use of the scenario defined weighting constants allows us to change the relative importance of the factors. For instance, in a given situation, the appropriate time on target is more important than the number of enemy killed or wounded. In other missions, the minimization of collateral damages to non-combatants might be the most important factor. Having the ability to adjust the pa-

[3] As discussed earlier, if we take into account the second and third order and non-kinetic effects often considered in joint operations, or most large service centric operations for that matter, the Lanchester equations are clearly unsuited and the details of the Evaluator will have to be adapted.

rameters allows us to guide the selection of the plans. In this prototype, we have chosen to implement a single set of parameters for the evaluation function at each level and for each of the elements. However, this is not a requirement, and each entity could have its own evaluation parameters. Additionally, since the Lanchester equations did not take into account the wounding of entities and there were no "Others" in the simple scenario both of these weighting factors were set to zero.

Results

One of our major concerns was that the variability of the OOS would mask any of the variability induced by the plans. However as shown in Table 3, this was not the case. As computational scientists, vice Subject Matter Experts (SMEs), our choices of parameter weightings might be sub-optimal, but they were deemed realistic.

Table 3. Relative Evaluated Values

Plan	Value
Direct Plan	1242.18
Minimum Cost / Preplanned Route	1187.24
Double Envelopment	1233.00

The use of the higher echelon TSPI XML files as way points for the route planning also proved successful. However, there were times when the routes were, at best, questionable. For example, after three successive double envelopments, the routes were rather cumbersome. An example of this is shown in Figure 9. The blue is the classic double envelopment. The orange lines represent the derived double envelopment decomposition of the double envelopment. From this it is easy to see that not all of the possible COAs are reasonable. However, just the fact they were generated was proof positive the architecture and process is valid.

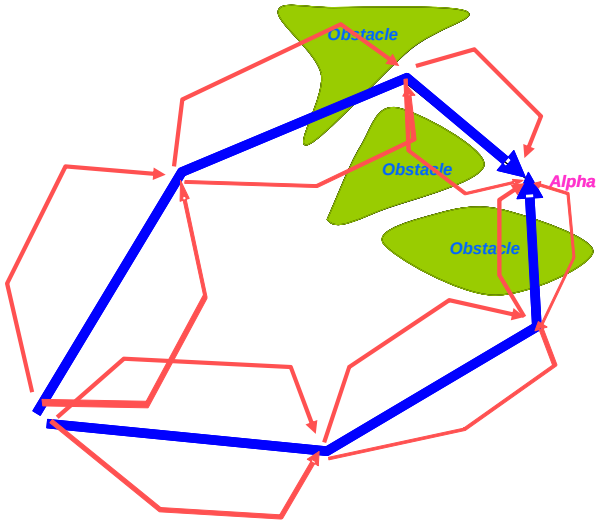

Figure 9. Overly Complex Generated Path

Conclusions

Like all proof of principles, the validity and applicability is in the eye of the beholder. Having said that, we feel we did proof the soundness of the basic architecture and approach, as well as the ability of the OOS system, with a few simple extensions, to support aggregate level models. The recommended solution, like many of the CBR based systems, often requires some "tweaking" to develop a near optimal plan. While the plans, in many cases, were clearly not ones developed by SMEs using deliberate planning, they were of sufficient validity and variability to support adjacent and opposing forces. In that context, we consider this proof of principle a success and the concept warrants further study. However, it is not ready for operational use.

Future Work

As the system stands right now, the process described in Figure 1 is largely a manual process. While we have ported the OOS and evaluation elements to a High Performance Computer (HPC) Linux Cluster (Powell at ARL), we have not automated the scripting and linkages between the elements. This will entail the construction of the batch job submission framework, a means of injecting different random number seeds for the OOS runs, and automated file management. While none of these steps are difficult in isolation, the sequencing and coordination between all the different processes and the files systems make this a logistical challenge. For example, OOS tends to exhibit "silent failures" when it runs out of resources. Since OOS shuts itself down, the system interprets this as a successful run. Thus, the issue of joining after all the runs are completed will produce errone-

ous responses. It is this type of nuance that makes the scripting of the system a challenging task.

Acknowledgements

This publication was made possible through support provided by Army Research Laboratory activities under contract number DAAD19-02-D-0001/Delivery Order 0644. The opinions expressed herein are those of the author(s) and do not necessarily reflect the views of ARL or the DoD.

References

Davis, Paul (1991) "The Base of Sand Problem", http://www.rand.org/pubs/notes/N3148/

Davis, Paul, (2005) "Introduction to Multiresolution, Multiperspective Modeling (MRMPM) and Exploratory Analysis", http://www.mors.org/meetings/tutorial/Davis.pdf

Pew et. al., Editors, (1998) "Modeling Human and Organizational Behavior: Application to Military Simulations," Panel on Modeling Human Behavior and Command Decision-Making: Representations for Military Simulations, National Research Council.

Pratt, David, et. al. (2006) "Progressive, Multi-Resolution Course of Action Analysis", 2006 Spring Simulation Interoperability Workshop, Huntsville, AL

Sidran, D. Ezra (2006) "The Lanchester Equation" http://www.cs.uiowa.edu/~dsidran/Lanchester%20Alternative.pdf

Tolk, Andreas, and Dietmar Kunde, (2003) "Decision Support Systems in the Military Environment", Chapter 6 in "Innovations in Decision Support Systems", Tonfoni G. and Jain L. (Eds.), International Series on Advanced Intelligence, Advanced Knowledge International, ISBN 0-86803-980-2, pp. 175-210, Magill, Adelaide, Australia, 2003

Tolk, Andreas: "An Agent-based Decision Support System Architecture for the Military Domain," Chapter 8 in Gloria E. Phillips-Wren and Lakhmi C. Jain (Eds.): "Intelligent Decision Support Systems in Agent-Mediated Environments," Volume 115 Frontiers in Artificial Intelligence and Applications, pp. 187-205, ISBN 1 58603 476 6, IOS Press, 2005

USMC Warfighting Laboratory, (2006)Project Albert web site, http://www.mcwl.quantico.usmc.mil/Albert/home.cfm