Collaborative Engineering Enterprise with Integrated Modeling Environment

Henry Ng, Suleyman Guleyupoglu, Frank Segaria
Naval Research Laboratory
Washington, DC 20375

Brett Malone, Scott Woyak, Greg Salow
Phoenix Integration Inc.
Blacksburg, VA 24060

Keywords:
Simulation Based Acquisition (SBA), Collaborative Engineering Enterprise (CEE), High Level Architecture (HLA), Integrated Model Environment, Virtual Enterprise.

ABSTRACT: *The U.S. Department of Defense (DOD) vision is to have a Simulation Based Acquisition (SBA) process that is enabled by robust, collaborative use of simulation technology that is integrated across acquisition phases and programs. The objectives are to reduce the time, resources and risk associated with the acquisition process. In the authors' experience a Collaborative Engineering Enterprise (CEE) is ideal to support the execution of SBA applications. CEE ties together the enterprise resources, project and engineering processes, and personnel needed for engineering development. CEE applies state-of-the-art information management, process workflow management, distributed computing, and modeling and simulation (M&S) technologies to provide a common framework within which the organization and its partners can collaborate. Phoenix Integration Inc. has developed an integrated simulation environment called ModelCenter® that enables a distributed team of engineers to perform engineering level modeling and analysis. It is well suited for supporting "System of Systems" engineering trade-off and optimization. In 2002, a Cooperative Research And Development Agreement (CRADA) between NRL and Phoenix was established. The goal of this agreement is to exploit recent advances in information technology to create an architecture that integrates task and management process, workflow, data and M&S services with both product models and simulations for supporting collaborative engineering enterprise. In this paper, we will give a brief overview of the CEE, process flow and integrated model environment, describe the objectives and motivation of our CRADA, the critical factors that drive the development, and design philosophy.*

1. Background

The U.S. DOD vision is to have a Simulation Based Acquisition (SBA) process that is enabled by robust, collaborative use of simulation technology that is integrated across acquisition phases and programs [1]. The objectives are to reduce the time, resources and risk associated with the acquisition process. A Collaborative Engineering Enterprise (CEE) can be used as a valuable tool to support the execution of SBA applications. With this motivation, the U.S. Naval Research Laboratory (NRL) developed a prototype CEE system in 2002 [2]. This system ties together the enterprise resources, project processes, engineering processes, and personnel needed for engineering development. CEE applies state-of-the-art information management, process workflow management, distributed computing, and M&S technologies to provide a common framework for the organization and its partners. This framework may be used to collaborate across different engineering domains and life cycle phases.

Phoenix Integration Inc. (Phoenix) has developed an integrated simulation environment called ModelCenter® that enables a distributed team of engineers to perform engineering level modeling and analysis. Users can quickly create an engineering workflow and then perform complex design exploration techniques to find the best solution. It can be used to link multiple simulations, cost models, and information technology models to create a powerful decision support environment. It automates the process of running hundreds of design programs that engineers need during a typical design project. It is well suited for supporting "System of Systems" engineering trade-off and optimization.

In 2002, a Cooperative Research and Development Agreement (CRADA) was established between NRL and Phoenix. The purpose of the CRADA was to create an architecture that integrates task and management process, workflow, data and M&S services with both product models and simulations for supporting collaborative engineering enterprise. In addition, an Application Program Interface (API) will be developed which can be used to integrate a standalone integrated simulation environment and applications into the CEE system architecture. This architecture may provide an integrated enterprise environment to support SBA applications.

2. CEE Objectives

The objective for CEE development is to establish a collaboration framework using existing or emerging domain specific resources to support scientific and engineering collaboration between distributed government and industry teams. When enterprises form a team to design and build a complex system, one of the first tasks that must be performed is the exchange of information. This information exchange is required for specification generation, document review, design, and system performance evaluation. Complex projects are usually executed by multi-skilled teams, whose members are often made up of personnel from both inside and outside the organization. Coordinating a complex project across the country or even around the globe is common. Resources and status reports should be available whenever they are needed, not just during traditional business hours. Collaboration across time zones can be critical to the success of the project. Our CEE vision is a system that electronically links government and industry partners who are members of a multi-tiered enterprise. Going far beyond a simple integrated e-mail and Web-based system, a CEE can be used to distribute and manage documentation and data associated with a large scale enterprise, serve as an on-line meeting place for team collaboration and tasking as well as providing common tools and management support capabilities.

The CEE provides two separate views for accessing tools and information. The first view is the personal view, which is primarily used by individuals to manage their activities. Each CEE user has only one personal view. The second view is the team/project-oriented view, which is used for information sharing among team members or different Integrated Product Teams (IPT). A CEE user can be part of many different team views. The team view allows members to collaborate both concurrently and non-concurrently.

A virtual personal office paradigm is used for the personal view (See Figure 1). When a user logs into the system, she is placed in her virtual office with typical tools that she would find in her physical office space. A status window is automatically displayed to enable this user to manage her tasks and activities for the teams and projects that she is involved with. The status window can show objects such as action items, notifications, reminders, chat request, and scheduled messages directed specifically to an individual.

For the team/project-oriented view, the paradigm of a "building, floor, and room" is used to provide the representations of the team collaborative space (Figure 2). This paradigm provides a persistent virtual space within which applications, documents, and people are directly accessible. It also serves as an on-line meeting place to support the team in accomplishing a particular task.

Defining rooms as the basis for communication means that users are not required to set up sessions or know physical user locations. Users need only to enter a room in the CEE environment in order to communicate with other team members. If users choose to communicate through audio, video or text, then the communication session is established automatically for them.

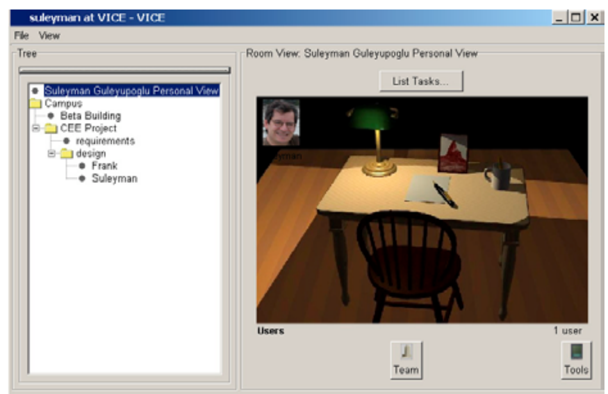

Figure 1: Personal View

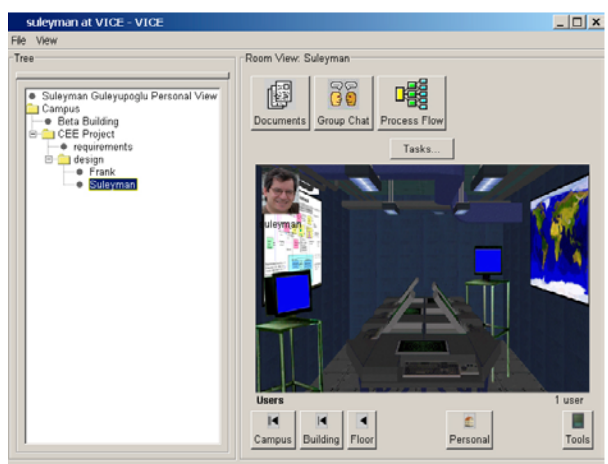

Figure 2: Team View

Basic collaborative tools, such as chat and white board, are provided in the CEE. The virtual room is also used as a place to provide access to generic or team-special tools that can be specified by users. One of these tools developed to increase productivity of teams is Process Flow concept and the editing tool.

Process Flow

One of the major difficulties in IPT operations is the challenge to communicate and work together efficiently because of the distributed nature of the team members in addition to their different disciplines, levels of expertise, and corporate culture. For this reason, a Process Flow capability was developed to aid team operations. The Process Flow application enables the members of the IPT to collaborate, define and develop the process to accomplish a given task. More importantly, it is used as a vehicle to foster communication within the team so that consensus can be reached on how to complete the task at hand early on in order to minimize miscommunication regarding individual roles and responsibilities for each team member. Figure 3 shows an example and the graphical interface of the Process Flow.

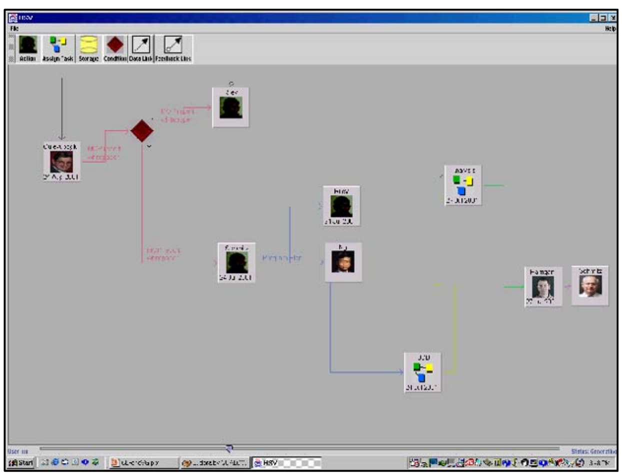

Figure 3: Example Process Flow Diagram

Process Flow is different from the traditional workflow diagrams in that it focuses on individuals' responsibilities. The intention of the Process Flow as a planning tool is to help identify who needs to do what in a team to achieve the common goal, which we refer to as a "task". It also helps monitor the progress made during the performance of a task.

An action item or unit of work assigned to an individual in the Process Flow is called an "action." An action typically requires input from others in the team and produces output for others to consume. Actions are represented as rectangular icons on the Process Flow diagram. The flow of data or sequence of events is represented with arrows connecting actions. A combination of connected actions makes up a task.

An appropriate member of the team "executes" the task after the planning phase is over and an agreement is reached on how to perform the task. This is shown graphically by illustrating the task as a process flow. Consequently, the CEE sends a notification to everyone in the team informing them that work on this particular task been initiated. As individuals complete their work (achieve the goal of their assigned action), they use a mechanism provided by the software to submit the product of their work to the system. These submissions are archived by the system at a central database and a reference to them is forwarded to the next person who needs to use the information to complete her task. Notifications will be automatically sent to each member of the team to inform them of the progress. At the same time, the icons representing the actions being taken change color to indicate the status. For example, action-in-progress status is indicated with blue icons, while action-completed status is represented as green icons. When the deadline of an action has passed, the icon turns red and the appropriate individuals are notified. It is also possible to set up the system so that individuals receive reminder notifications as deadlines on their assignments approach.

3. Integrated Model Environment

The Integrated Model Environment (IME) enables engineers to perform distributed modeling and analysis. The IME is a tool for automating and integrating engineering analyses. It can greatly decrease the turnaround time needed for conducting trade studies. The IME enables users to incorporate analysis software or simulations into reusable components that can be published on the local or wide area networks. Once published, users can access these components from any computer on the network through a standard protocol. Simulation codes are wrapped and maintained to run on their native platforms: UNIX or Microsoft Windows. Part of the attractiveness of the IME concept is to allow anyone, anywhere to access sophisticated analytical models on their own PC at their own desk. The IME includes three elements: simulation component wrapping; distributed model selection & linking; and system selection & optimization algorithms.

Simulation Component Wrapping

The first step in creating the IME capability is to be able to encapsulate each simulation or analysis tool in a reusable IME component so that a common client can interface the key parameters. IME components can be created from such dissimilar commercial off-the-shelf (COTS) applications as Matlab®, Excel®, or legacy tools owned by individual parties. The "wrapper" is used to parse information and expose that information in a common API. See Figure 4.

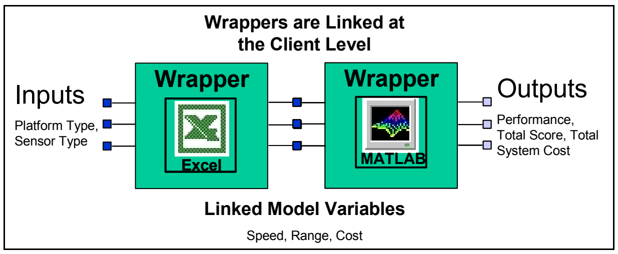

Figure 4: Linking Applications

Distributed Model Selection & Linking

A graphical integration interface is used to select the distributed, reusable IME components and link them together to form a single integrated model. The Distributed Model Selection & Linking module is also used to link and translate data and share information between all models in the system.

System Selection and Optimization Algorithms

Numerical search routines are then used to drive the integrated model to explore the design space effectively. Simple tools such as Design of Experiments can be used to set up trade space runs while optimization routines are called upon to pinpoint the best design based on constraints. In addition, different third-party optimization algorithms can be selected and embedded in the IME by using a set of APIs.

Phoenix's ModelCenter brings legacy software and commercial tools into an open, integrated environment. With an entire tool set of analyses working together as modules inside of a single application, users can combine these tools to solve a variety of problems.

ModelCenter uses unique integration architecture with another module called "Analysis Server®" to wrap and integrate legacy programs, data, and geometry features. Using Analysis Server, designers can access multiple design programs, databases, and simulation APIs from remote computers. ModelCenter and the Analysis Server provide a client/server environment. Components are created and distributed using the Analysis Server and then integrated using ModelCenter. ModelCenter provides tools such as optimization drivers and is integrated with standard object protocols such as COM. Figure 5 shows an example set of engineering models linked with a design concept within ModelCenter.

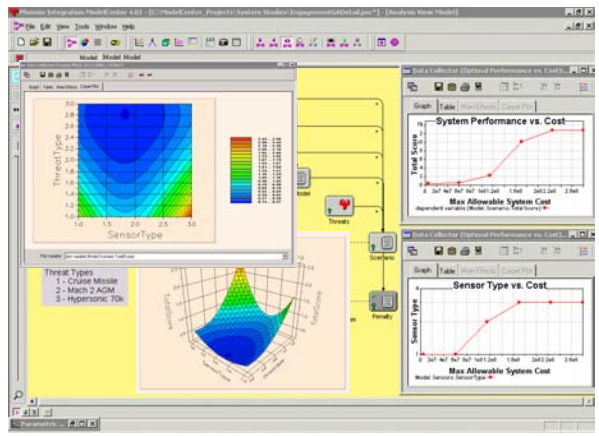

Figure 5: ModelCenter Interface

ModelCenter contains a gradient-based optimization algorithm that can be used with any model or combination of models in the system. The optimization methods are simple and will allow designers to develop constrained problems and search for the best possible design. Depending on the fidelity and the computational expense of the models, ModelCenter's optimization routines can quickly search through the design space and find optimal parameters. In addition, the users can embed external optimization algorithms using a set of ModelCenter APIs. Using the optimization tool, users can select an objective function from one model and use the output parameters from another model as a constraint set. Using an IME such as ModelCenter, stand-alone legacy programs from distributed computing resources can work together as if they were modules of a single analysis program.

4. Integrated Model Environment with Process Flow

The major area of collaboration between NRL and Phoenix is the embedding of the IME capabilities into the Process Flow application. A wrapper, or interface, is being developed to allow data to pass transparently between selected actions in Process Flow and ModelCenter models, thereby enabling IME software to be used as part of the CEE. The advantage of integrating the IME into the Process Flow would enable us to streamline the task execution end to end, potentially enabling personnel with different backgrounds to interact more efficiently.

In order to provide an example of the integration, suppose a team is investigating future system needs in support of network-centric warfare. Using the Process Flow application tool, the distributed team formulates how this task will be accomplished. In the task formulation phase, the team starts the Process Flow application in CEE to work together to determine *what needs to be done* and, when appropriate, *who needs to do it* in a collaborative environment. The Process Flow representation of this task is generated and is shown in Figure 6.

In the first box on the left, Mr. Segaria formulates an analysis plan, which stated the objectives of the investigation. The analysis plan is sent to Mr. Stumm (as shown in the second box on the left side) to review and for approval. If the analysis plan is not approved, it is sent back to Mr. Segaria for revision. This review and approval cycle is repeated until the analysis plan is approved. After the analysis plan is approved, it is forwarded to the Wargaming Group to conduct a war game exercise using a HLA Federation to evaluate the combat performance. This is shown as a sub-task icon, i.e. task within a task. Sub-tasks are used to group tasks into smaller portions or to delegate planning of sub-tasks to others.

The wargaming results are then fed to an Analysis Group to identify the shortfalls and future capability requirements. These future capability requirements are sent to an Advanced System Concept Formulation Group to identity a list of candidate systems, which can address these needs. These candidate systems are fed to a Scenario Generation Group to formulate a scenario that can be used to conduct system trade-off and optimization as well as to set up the input files needed for each individual system component model to be used for the trade-off study. Having completed the scenario and having set up individual system component model input files, the System Modeling Team conducts system evaluation and trade-off to determine the most cost effective solution using the IME (ModelCenter in this example). When the task

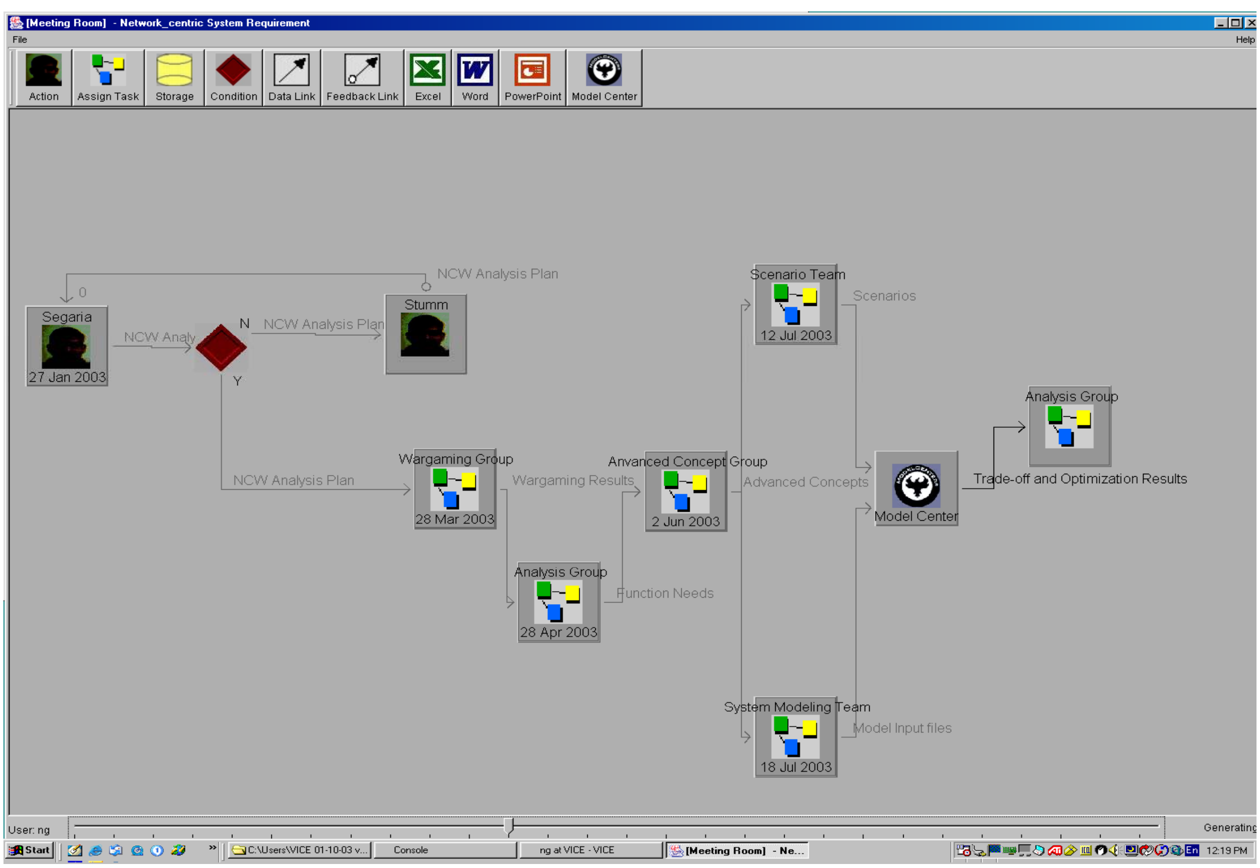

Figure 6: Process Flow Example

execution reaches the IME box, the system trade-off and optimization setup in the IME will be automatically executed without human interactions. Trade-off study results will be generated.

An example problem would be three different sensors, three different platforms, and three different threats to be considered for the system trade-off. In constructing a design environment to evaluate candidate concepts for network centric systems, the impact of cost, system performance and variability of threats must be considered. The system designer must have the ability to add subsystems, or "nodes," to the battlefield network, increasing the complexity of the problem and requiring more sophisticated search algorithms to analyze thousands of combinations of each subsystem design. Since each sensor and platform has associated cost, we have an additional analysis component that computes total system cost. The cost is used as a constraint by the Genetic Algorithm [4] optimizer. A graphical depiction of the example engagement is shown in Figure 7 below.

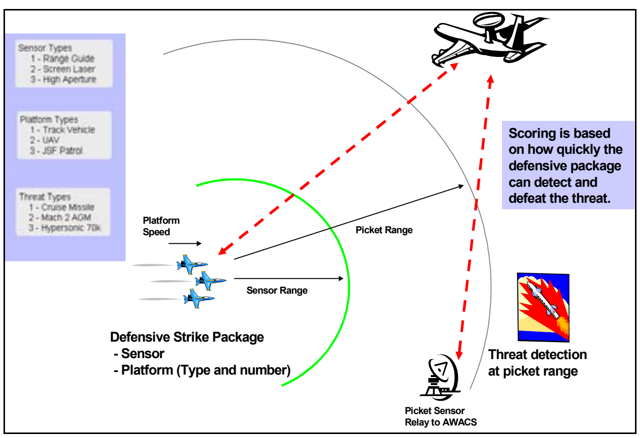

Figure 7: Example Engagement Scenario

When the integration of the Process Flow and IME is complete, the user will be able to click on the ModelCenter box to initiate trade-off analysis. Using ModelCenter, shown in Figure 8, each of the three main applications (Platforms, Sensors, and Threats) are integrated into a scenario model. There are three input sources (Platforms, Sensors, and Threats). The Defense System model combines the platform and sensor to form an overall system. That information is fed into the Cost Model and the Scenario model.

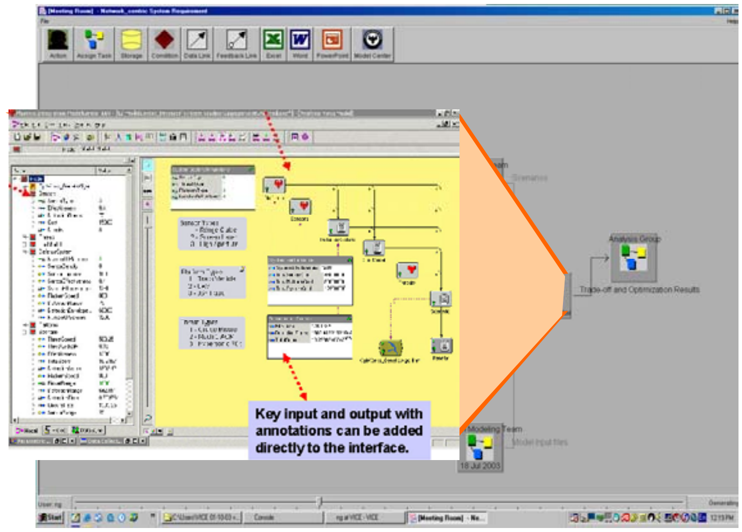

Figure 8: Integrated set of models in ModelCenter with optimization feedback.

The threat model is also linked to the Scenario, which calculates the total score based on the defense system's ability to defeat the threat.
A penalty function is used to compute the figure of merit for the Genetic Algorithm to find the best combination of sensor and platform to maximize the total score while maintaining cost within budget.

Trade-off Studies

Basic system trades to understand the nature of each component will be generated automatically as setup in the IME box in the CEE Process Flow task. Figure 9 shows an example of a contour as different sensors are played against different threats. The score is the overall effectiveness. Carpet Plot tool in the ModelCenter will be used to run multiple threats and sensors automatically.

By using maximum cost as a variable and optimizing at each point, the effect increasing total budget has on system performance will be easily seen and available to the analysis team to evaluate. For example, each point (see Figure 10) represents an optimal solution given the max cost specified in the *x*-axis. Each design uses a different set of sensors and platforms selected by the optimizer. When the cost limits are relaxed, the optimizer is free to choose components that perform better. This chart is for one fixed threat.

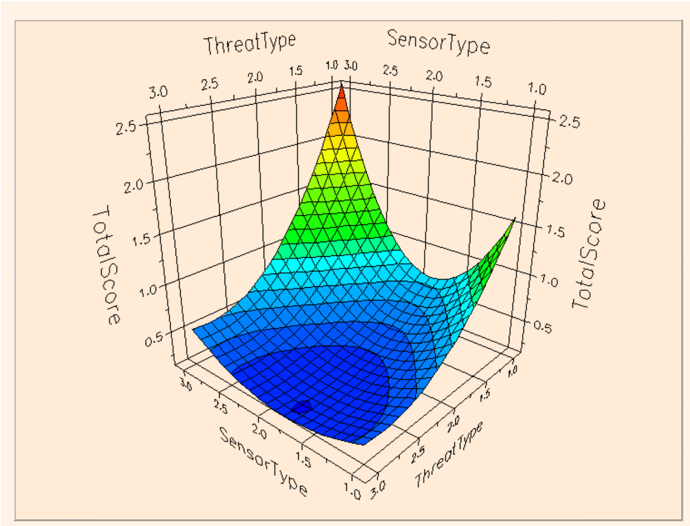

Figure 9: Assessment of Total Score given sensor type and threat type.

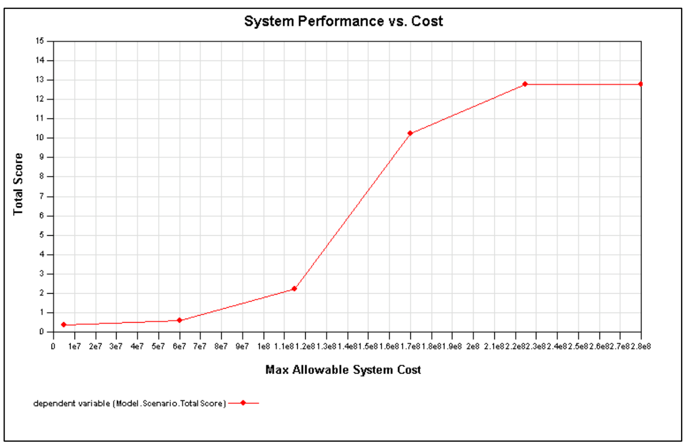

Figure 10: CAIV Study shows optimal system selection. Each point is optimized solution with the cost as a constraint.

Figure 11 shows the optimal solutions for Cost As an Independent Variable (CAIV) studies with different operational threats. The effect of different threats to the variable budget can be used to build a surface. This trade shows a surface plot of the optimal sensor/platform package for a given threat and cost constraint. Each point is optimized to maximize score within the specified budget (cost constraint). The chart shows that given unlimited cost against a low-order threat, a high score can be achieved. Conversely for high threats and low cost, low scores are achieved. Also, there is a point where increasing the cost budget does not improve score because the technology becomes limiting.

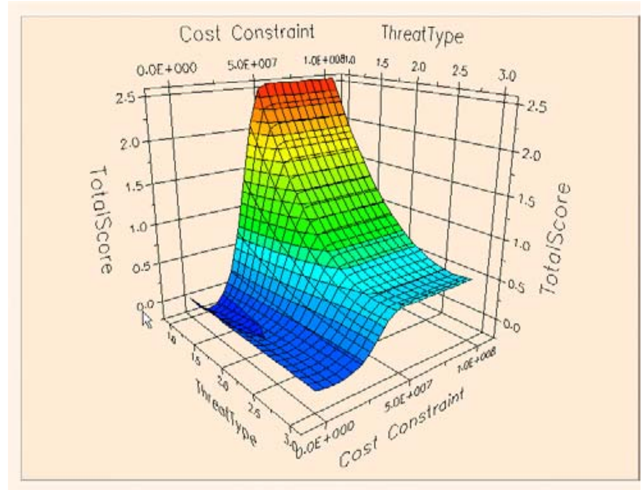

Figure 11: Optimal solutions for CAIV studies with different operational threats.

These types of results can be generated automatically according to the setup of the IME. The results are stored in the CEE. Anyone in the team as authorized can find the information very easily. Changes to input can be made very easily and the results can be re-generated in a very short time. This kind of integrated capability allows users to have a better understanding of the results and find the optimal solutions in a timely manner.

5. CEE and IME Challenges

There are certain challenges that will need to be addressed in integrating the two systems from NRL and Phoenix. These challenges are not unique to these systems however. Creating a generic environment where simulation data is passed from individuals to an automated trade study system will require significant forethought. In the most basic sense, the CEE system needs to be able to pass data to the IME for processing. This can be accomplished in several ways. If both systems are running on the same computer, using the file system is a trivial solution. However, in most cases, the more appropriate solution is to have the CEE server act as a central data repository where simulations can access the input data and store their results. This would allow users of CEE to access the data easily even if they do not have access to the IME system directly.

IME system shallprovide a set of API that allows CEE to pass data to it programmatically. These APIs will include functions to initiate the IME with configuration files that will set up the IME environment. Another API will allow CEE to execute IME trade studies in a closed loop. Finally, CEE will be able to request that the IME store its results at a certain location.

Another issue is consistent information representation. CEE Process Flow is a distributed application that allows a team of people at different locations to work together and interact with each other. As users make changes to a particular content, the updated information is displayed at all on-line participants' displays to reflect the change so the group can have a consistent picture. It will be more of a challenge to do so with the IME, which in our case is ModelCenter. ModelCenter GUI is developed as a single user system even though it can work with distributed simulations to perform trade studies. It will need distributed display capabilities for distributed team operation.

6. Summary

The incorporation of the IME in CEE will provide the capability to integrate the system engineering process with models and simulations seamlessly. The demands of SBA have put a new emphasis on engineering modeling and integration. CEE with IME framework would enable the distributed team to interact more effectively by not only creating an environment, where it is easier to access information, but also easy to generate information. System engineers can use a rapid modeling environment, and sophisticated analysis control tools to investigate the concept design space. Using the features of the optimization module and the parametric trade study tool in the IME, engineers can gain more insight into the concept model. This in turn supports SBA activities by providing quantifiable benefits of modeling and simulation as measured in: cost, schedule, productivity, quality and performance.

References

1. http://www.ntsc.navy.mil/Resources/Library/Acqguide/sba.htm (last visited on Apr, 1, 2003)
2. Schmorrow, D.; Riley, B.; Ng, H.; Guleyupoglu, S.; Segaria, F. "Collaborative Engineering Enterprise for WARCON", Simulation Interoperability Workshop, 02S-SIW-080, March. 2002.
3. Malone, B., Papay, M, "ModelCenter: An Integration Environment for Simulation Based Design," Simulation Interoperability Workshop, Orlando, FL, Mar., 1999.
4. Goldberg, D. E.; "Genetic Algorithms in Search, Optimization and Machine Learning," Addison-Wesley, Reading, MA, 1989
5. Malone, B. "A Simulation Based Design Environment for Network Centric Warfare Using Distributed Analysis," DASD 33, Advanced Simulation Technologies Conference, Orlando, FL, 2003.

.

Author Biographies

HENRY NG is the Head of the Distributed, Virtual Collaborative Environments Section of the Advanced Information Technology branch in Naval Research Laboratory. He has been actively involved in simulation and modeling over twenty years. Prior to joining NRL, he was the Head of the Simulation and Modeling branch of the Warfare Analysis department of NSWCDD in White Oak, Maryland. He was the architect of a large-scale sea, space, and land battle force level simulation model known as MARS (Multi-warfare Assessment and Research System). Henry was a member of the Technical Support Team to support DMSO Architecture Management Group (AMG) to develop High Level Architecture during 1994 –1996. He was awarded with a Navy Meritorious Civilian Service Award for his contribution in Modeling & Advanced Distributed Simulation in December 1994.

SULEYMAN GULEYUPOGLU is a research scientist of ITT Industries, Inc., working at the U.S. Naval Research Laboratory as an on-site contractor. He received his Ph.D. degree in Engineering Science and Mechanics from The University of Alabama, Tuscaloosa, AL. His current research interests include distributed simulations, virtual reality, collaborative engineering, and expert systems.

FRANK SEGARIA is a computer scientist at the Advanced Information Technology Branch at NRL. Frank has extensive experience in the modeling and simulation, mission planning and software engineering. For the past two years, Mr. Segaria has conducted performance assessment on the HLA RTI in supporting of Naval applications for the Navy Modeling and Simulation Program Office (N6M). Specifically, he was responsible for the design, develop, and testing of an Aggregation/Deaggregation minefield Federation. This allows different abstraction levels of simulations to interoperate in joint simulations. In addition, he was also part of the Consistent Network Information Stream (CNIS) team, which used agent technology to develop a system to gather information for many different sources to provide a consistent tactical picture for different level of commands. Mr. Segaria received a B.S in Computer Science from the University of Maryland in 1993.

BRETT MALONE is the co-founder and President of Phoenix Integration located in Blacksburg, VA. Dr. Malone holds a Ph.D. in Aerospace Engineering from Virginia Tech and he focuses on operations research and system optimization and works with top industry customers on improving their engineering process through software integration. Through his management, Phoenix has successfully developed business in the US, Asia, and Europe. Dr. Malone routinely coordinates with other industry partners and research institutions to form partnerships for successful product development projects. Dr. Malone's background is in engineering design and optimization for the aerospace industry. He has experience in the planning and management of large development projects through his experience at Phoenix Integration, as a research associate at Virginia Tech, and as a project leader for the ACSYNT Institute. Prior research background includes free-flight optimization for NASA Ames, development of optimization control environment for the ACSYNT Institute, HSCT concept optimization, and development of MDO algorithms for optimizing subsonic transport designs. Areas of investigation include genetic algorithms, parametric multi-objective optimization, and numerical mapping techniques for visualization of n-dimensional data sets.

SCOTT WOYAK is co-founder and Director of Software Solutions of Phoenix Integration. Dr. Woyak holds a Ph.D. in Mechanical Engineering from Virginia Tech and formed Phoenix Integration based on research performed during his studies. He has led software development efforts including ModelCenter, Analysis Server, and various research related contracts for Phoenix Integration. Dr. Woyak now works with industry to help build and deploy software frameworks for engineering design. Areas of research include human factors in engineering software, automation and integration techniques for engineering analyses, distributed computing, and design exploration techniques including optimization and probabilistic design techniques.

GREG SALOW serves in a business development capacity for Phoenix Integration's aerospace and defense market group. He has industry experience with engineering, mathematical, and visualization software organizations such as The MathWorks and CLEAR Software. He received a B.A. degree in Communications and Economics from Western Washington University.