2006 CCRTS

THE STATE OF THE ART AND THE STATE OF THE PRACTICE

AN ANTICIPATORY ENVIRONMENT FRAMEWORK

C2 Architecture
C2 Analysis
C2 Experimentation

Authors:
Steve P. Colenzo
William E. McKeever
Chad C. DeStefano
Duane A. Gilmour

POC: William E. McKeever
Organization: Air Force Research Laboratory, IF
Address: Air Force Research Laboratory, IFTC
Bldg 3/ Room E-7
525 Brooks Road
Rome Research Site
Rome, NY 13441
315-330-2897 / 315-330-2953
William.McKeever@rl.af.mil

Report Documentation Page

Form Approved
OMB No. 0704-0188

Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.

1. REPORT DATE **JUN 2006**	2. REPORT TYPE	3. DATES COVERED **00-00-2006 to 00-00-2006**
4. TITLE AND SUBTITLE **An Anticipatory Environment Framework**		5a. CONTRACT NUMBER
		5b. GRANT NUMBER
		5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)		5d. PROJECT NUMBER
		5e. TASK NUMBER
		5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Air Force Research Laboratory,AFRL/IFTC,525 Brooks Road BG 3/Room E-7,Rome,NY,13441**		8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)		10. SPONSOR/MONITOR’S ACRONYM(S)
		11. SPONSOR/MONITOR’S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES
The original document contains color images.

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**		**15**	

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std Z39-18

An Anticipatory Environment Framework

Stephen P. Colenzo, William E. McKeever, Chad C. DeStefano, Duane A. Gilmour
Stephen.Colenzo@rl.af.mil, William.McKeever@rl.af.mil, Chad.DeStefano@rl.af.mil, Duane.Gilmour@rl.af.mil

Air Force Research Laboratory
Information Directorate
525 Brooks Road, Rome, NY 13441-4505

Abstract

In today's volatile world, it is necessary to provide decision makers with every advantage when dealing with a dynamic and changing adversary. Decision makers require a capability that would enable them to anticipate and shape the battlespace, i.e. an anticipatory environment (AE). This capability would lead to more proactive, vice reactive, decision making in future military missions. An AE proof of concept framework has been developed that encompasses the first four phases of the Joint Air Estimate Process (JAEP), as defined in Joint Publication 3-30, 'Command and Control for Joint Air Operations' [1]. The framework provides the foundation to perform mission analysis, situation and course of action (COA) development, COA analysis, COA comparison, and aids in COA selection. This paper presents the tools, technologies and the integration of the necessary capabilities to create an AE. Utilizing a simple scenario, containing friendly and adversary entities, the AE framework will be demonstrated along with a discussion of the results.

Keywords: Anticipatory Environment, EBO, Predictive Battlespace Awareness, Simulation, Mission Planning

1.0 Introduction

The military planning process depends upon analysis systems to prepare for the battlespace. Complex technical challenges exist in developing automated processes to derive hypotheses about future alternatives for mission scenarios. The military conducts combat operations in the presence of uncertainty and the alternatives that might emerge. It is virtually impossible to identify or predict the specific details of what might transpire. As a result, the military planning process must continuously be updated/modified to account for the latest adversary actions. Consequently, the military is often reacting to the latest adversary actions instead of anticipating the future.

The premise of an anticipatory environment is an interactive environment that would enhance the decision maker's ability to anticipate, shape and dominate the future battlespace. This would lead to more proactive decisions and resultant actions in the presence of a dynamic changing adversary, ultimately moving them towards the desired end state. Anticipation would provide decision makers with the ability to foresee the battlespace resulting in optimized decisions based on resources, constraints, and time available. An AE would provide recommended actions, and the appropriate times and places to perform them, to achieve the desired affects on the adversary to constantly shape the battlespace. Combined, the ability to anticipate and shape would enable a capability to get inside our adversary's decision loop and generate plans that ultimately lead to dominance in the battlespace. This would enable a savings in manpower and resources as potentially dangerous situations could be avoided. Some of the desired characteristics of an AE are as follows: (1) high-fidelity models (red, gray & blue) are dynamically produced and updated; (2) many candidate courses of action (COAs) are automatically produced and continuously evaluated; and (3) simulations are conjoined with live operations for dynamic situational assessment and prediction. The ability to achieve these desired characteristics and the overarching AE would result in planning staffs having a better understanding of the mission space (past, present & future) and result in the generation of plan(s)/options that would "virtually checkmate" the adversary. Decision makers could ultimately answer these questions: can I anticipate the adversaries next move, and how can I use this anticipation to my strategic advantage?

2.0 Background

Current technology and approaches to support an AE have fallen short. The models currently utilized to represent red, gray and blue actors are static, low fidelity and fail to capture realistic behaviors. A capability is needed to model individuals and groups with high fidelity to incorporate intent, goals, actions, as well as anticipation of realistic and unexpected behaviors. The current COA development process is predominantly manual, and takes hours, if not days to fully develop a limited number of COAs. To support an AE, a capability to automatically generate candidate COAs that utilize different approaches to achieve commander's intent and also take into account a dynamic adversary must be developed. A candidate COA is one that is suitable as well as feasible. A COA is suitable if it is in alignment with commander's intent and will accomplish the mission when carried out successfully and a COA is feasible if it can be achieved given the available resources. The COA development process must also account for all possible blue actions; including diplomatic, information, military and economic (DIME) instruments of national power.

The developed COAs must be analyzed dynamically; however, the current process used to analyze COAs is extremely manpower intensive and is generally accomplished utilizing teams that attempt to represent the action/reaction/counteraction nature of friendly and adversary forces in conflict. Also, this technique cannot be maintained at the speed of current operations. As a result, a limited number of COAs are analyzed dynamically. Currently, automated techniques, which have been developed and are faster than manual approaches, are performed utilizing a scripted adversary and focus on attrition based modeling. This prescripting fails to account for the dynamics of conflict. Also, they are incapable of assessing effects on the adversaries political, military, economic, social, information and infrastructure (PMESII) systems and their contribution to the overall mission objectives, which is inherent in effects based operations (EBO). A capability is needed to be able to dynamically model and reason over the adversary's complex system to synchronize all blue actions with effects on the adversary's systems as well as a capability to simulate all of these actions, i.e. kinetic, non-kinetic, indirect, complex, cascading and recovery events, much faster than real-time. An AE system would be able to support the human decision making process and allow them to evaluate an increased number, orders of magnitude, of candidate COAs rapidly, taking into account effects based approaches and a dynamic adversary.

The introduction of technologies to develop and evaluate an increased number of candidate COAs, leads to the necessity to develop a capability to automatically grade and evaluate COAs against objectives for rapid comparison and selection. Current techniques are highly subjective and won't support the capability to rapidly compare hundreds of disparate COAs. A more rigorous objective methodology is necessary for an anticipatory environment that allows for comparison of radically different simulated COAs that achieve commander's intent. Once a COA is selected and is being accomplished in operation, real-time data must be compared with simulation results to determine alignment with commander's intent. Decision makers must be able to answer the questions; is that planned action appropriate or do we need to re-evaluate? Currently, this is very difficult. An AE would be an "always on" capability; continuously assessing engagement result verses predictions, and developing and analyzing candidate COAs to move the campaign from its current state to the desired end state, when necessary.

To summarize, the following capabilities must be developed to support an AE; *capability to* (1) model individuals/groups (red, gray, blue) with high fidelity, (2) model & simulate effects, (3) automatically generate candidate COAs, (4) automatically grade/evaluate COAs against objectives, (5) support multiple parallel COA analysis and (6) continuously assess engagement results vs. predictions *faster than real-time*. The authors, with the support of contractual expertise have made significant strides towards the development of an AE. We present here a proof of concept of one such AE. This represents the "tip of the iceberg" of some of the required capabilities to support and fully develop an anticipatory environment. It currently encompasses the first four phases of the Joint Air Estimate Process (JAEP) [1]. The framework provides the foundation to perform mission analysis, situation and course of action (COA) development, COA analysis, COA comparison, and aids in the COA selection process. The balance of this paper presents the tools, technologies and the integration of the necessary capabilities to perform these four phases of the JAEP, as well as a demonstration of the framework utilizing a simple scenario.

3.0 AE Framework

The AE framework, which is illustrated in Figure 1, was implemented with recently developed tools and capabilities as well as those in current development. Figure 1 depicts the capabilities and their relationship to the phases of the JAEP, along with the flow between the capabilities. The JavaCOG tool, which was developed in-house, was used to perform mission analysis as well as COA development. Additional tools that were used for COA development were the Strategy Development Tool (SDT), developed by BAE Systems (formerly Alphatech); and the Scenario Generation (SGen) capability developed by Securboration. To perform COA analysis, the Force Structure Simulation (FSS) capability, which was originally developed by Metron, but now is enhanced and maintained in-house, was used. The COA Simulation Analysis (CASA) process developed by SAIC was used for the COA comparison phase. These tools are outlined in bold and are further discussed in sections 3.1 - 3.4. Since most of the tools were developed independently, significant effort was accomplished to develop data translators to integrate them into the AE framework. The data translators are discussed in section 4.

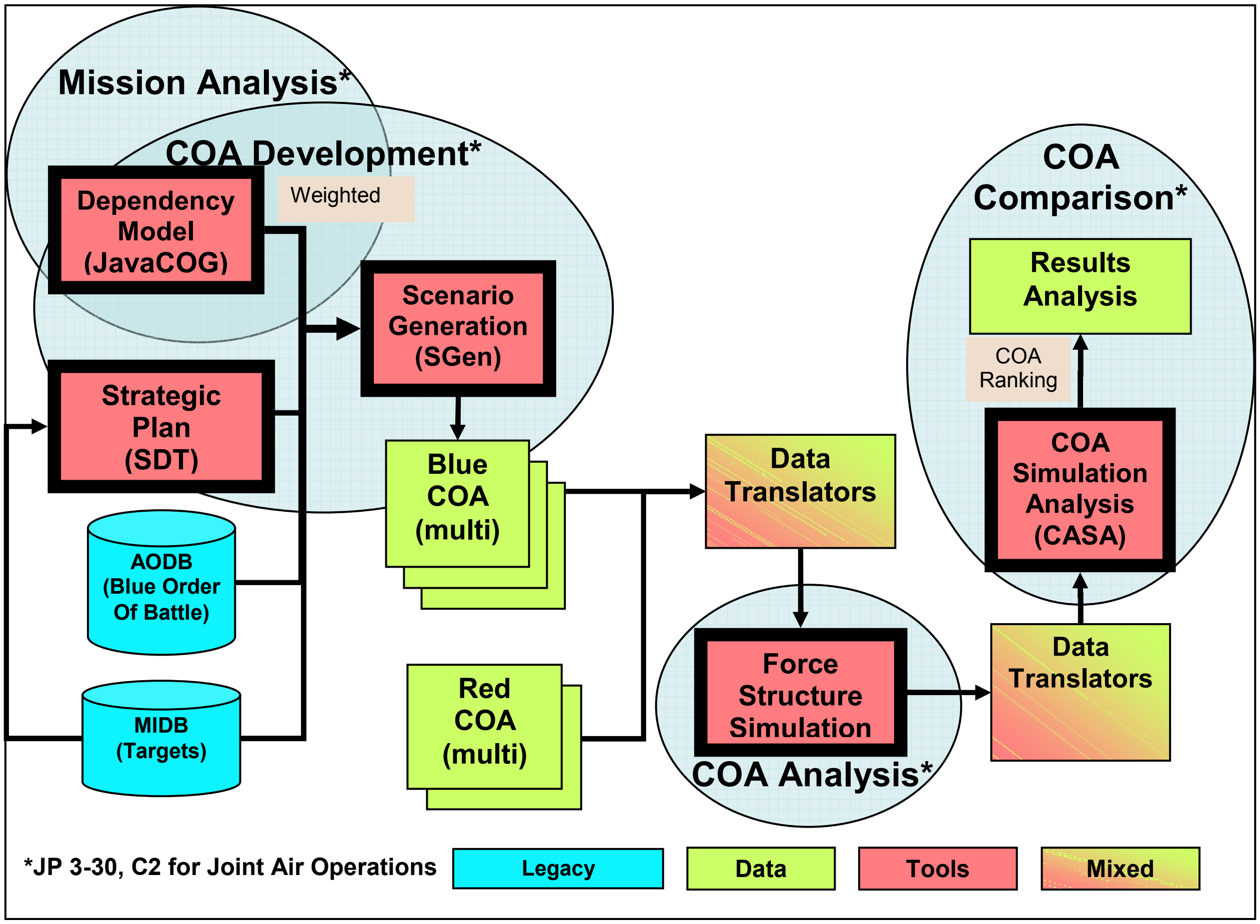


Figure 1. AE Framework Flow Chart

3.1 Mission Analysis

Mission analysis is the first phase of the JAEP. This phase focuses on "analyzing the joint force commander's mission and guidance to produce a joint air component mission statement" [1]. Intelligence preparation of the battlespace (IPB) is a major portion of this phase. One of the objectives of IPB is to identify adversary and friendly centers of gravity (COGs) at the strategic and operational levels. The COG modeling process is a critical aspect of effects based planning, simulation and assessment as it results in models that represent the links and nodes of the adversary's complex system. The IPB process, along with COG modeling, has traditionally been accomplished by humans at either reach back facilities, or within the air operations center (AOC). These human intensive and time consuming processes are not dynamic enough to support an AE. This is where tools can increase the reliability and effectiveness of the models. By providing experts with a more streamlined approach to create models, more complete models will be developed and the warfighter will have a better understanding of their actions which will result in the creation of more effective COAs. A tool called JavaCOG was utilized to perform COG modeling and is currently the starting point of our AE framework.

The choice to use JavaCOG was based on the support it provides to the other technologies in the AE framework. The COG relationships defined in JavaCOG can be imported into SGen to enable effects based planning. JavaCOG can be used to create a parameter file for FSS, which enables FSS to simulate effects, and also supports the CASA capability by identifying direct and indirect relationships. While JavaCOG is an effective tool for capturing the relationships and dependencies within the COG models and providing the models to the other tools, it does not provide the streamlined capability to reason over the models.

JavaCOG, a java-based COG authoring tool, is used to develop the nodes within the COGs and define their relationships. It provides a graphical technique to develop COG models and extends the model to include EBO characteristics. Furthermore, JavaCOG provides a flexible means to develop high abstract level COG models as well as more detailed lower level models.

JavaCOG's main window, shown in Figure 2, shows a graphical representation of a dependency model. The interconnections of the nodes in the model are represented with arrows, with each arrow pointing to a dependent node, e.g. in Figure 2, the *power grid* depends on *work force ID_1*, *power plant ID_1* and *power plant ID_2*.

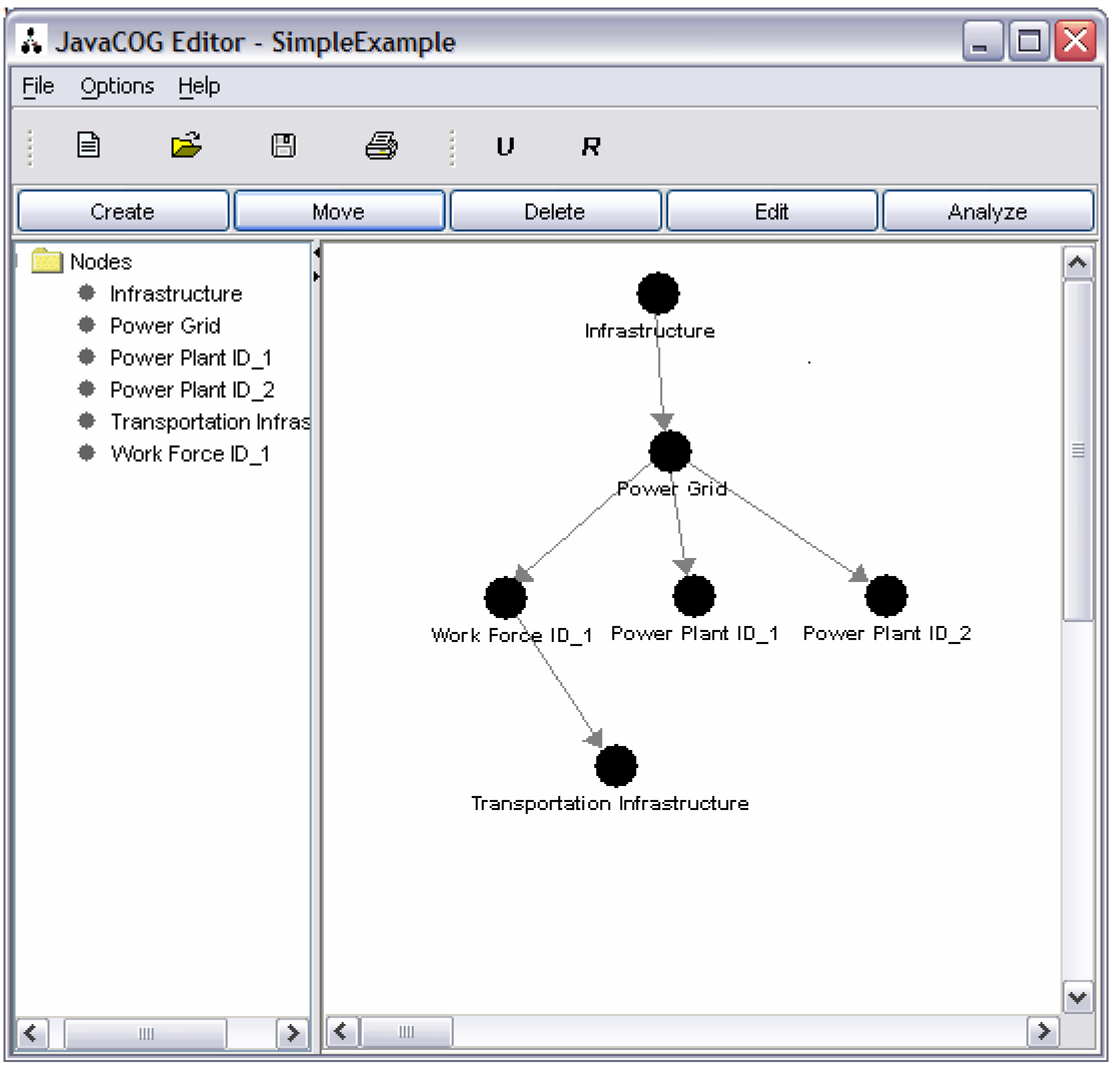


Figure 2. JavaCOG Main Window

The user can further define a node's properties by utilizing the editing mode. Figure 3 is an example of the screen a user would see when editing the *power grid* node. The top half of this screen allows the user to edit the *name* of the node, *reference / BE number*, *category code*, *repair time* and the *indicators* for this node. The *repair time* is an estimate of how long it will take for the node (*power grid*) to be repaired after a blue force action has affected the node. On the lower half of the screen are the properties of the nodes that can affect the *power grid*. The *influencing attribute* drop down menu allows the user to select a specific node to edit. The user can select the probability that an influencing node will cause the main node to be affected, e.g. in Figure 3, if the *power plant ID_2* node was affected, there would be a *very high* probability that the *power grid* node would be affected in a cascading event.

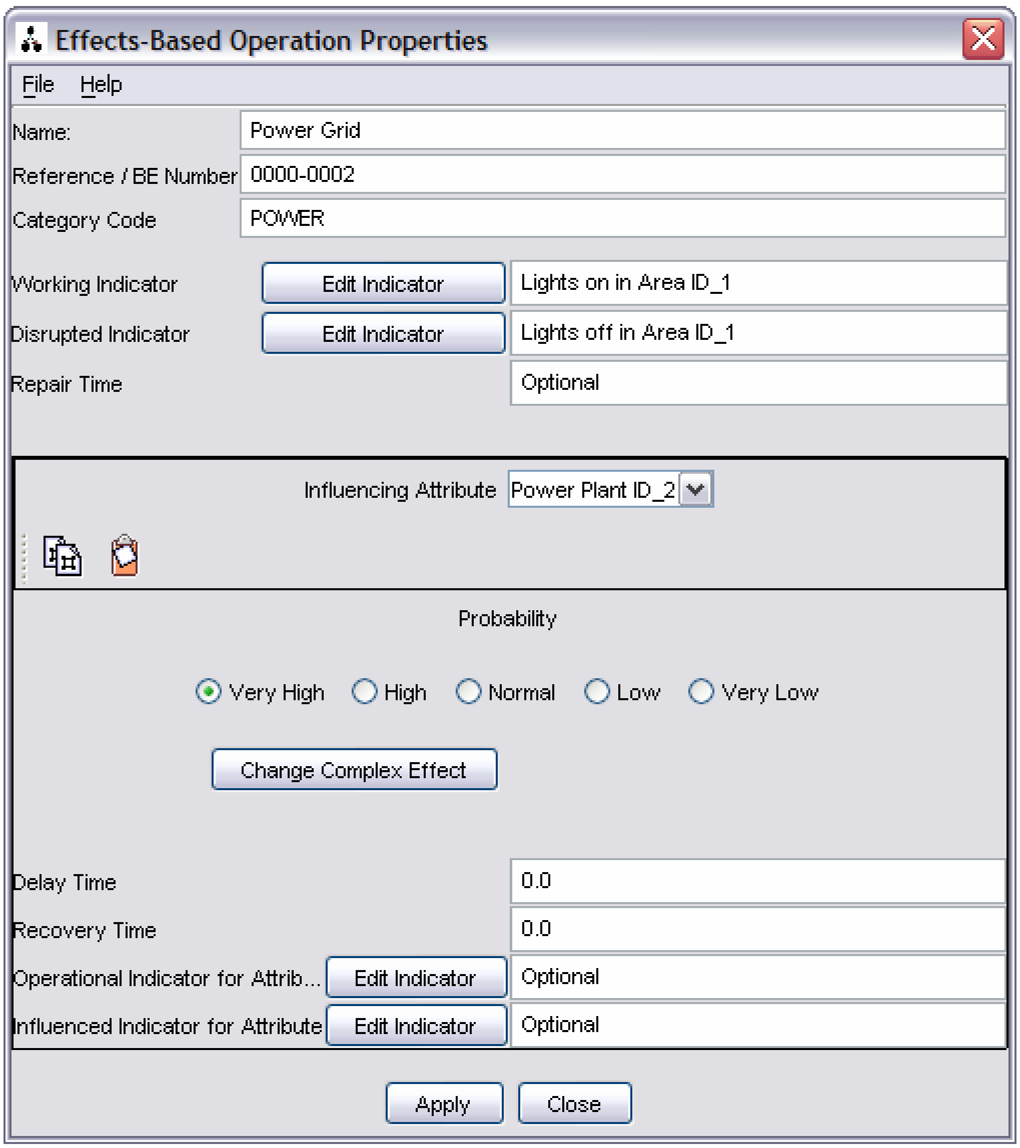


Figure 3. Editing EBO Properties in JavaCOG

Next, the user has the option to assign a complex effect for the influencing node. A complex effect is defined as an effect where some or all of the influencing nodes must be affected to influence the main node. In this example, referring to Figure 2, there could be a complex effect between the two *power plants* which will have the desired affect on the *power grid* node. In other words, both *power plants* would have to be degraded to affect the *power grid*. To modify the complex effect, the user clicks on the *change complex effect* button which brings up a window of all possible complex effects in a check box grid. The user will then check off the nodes that make up the complex effect. The next property that can be edited is the *delay time*, which is the amount of time it takes for an effect on the child node to influence the main node. The *recovery time* is the amount of time it takes for the recovery event on the child node to affect the main node. The last properties to be edited are the *indicators*. The *indicators* in this section are in reference to the observations that can be made at the main node (*power grid*) that would indicate the state of the influencing node (*power plant ID_2*).

3.2 Situation and COA Development

The next phase of the JAEP is situation and COA development. The purpose of this phase is to further refine IPB that was accomplished during mission analysis and perform COG analysis. In addition, multiple blue COAs are developed. The proposed framework uses two technologies, in addition to further developing the COG model with JavaCOG. The first technology, the SDT [2], was used to develop a high level COA. The SGen [3] technology was used to develop more detailed lower level COAs for simulation and analysis.

The COA development process is a highly manpower intensive and time consuming process. The SDT was developed by BAE systems to shorten this process. It helps users, through the use of templates; to develop effects based COAs, which was its application within the AE framework. In order to develop this initial COA, knowledge from Subject Matter Experts as well as Air Force doctrine was utilized. Even with the help of the SDT, the process of developing an effects based COA was a timely and manually expensive operation.

A major concept behind an AE is the ability to rapidly generate relevant COAs. Current scenario generation is extremely labor intensive, and often takes several man days to months to develop a few COAs. The process of generating the required scenario files is manual and requires a person or persons to collect requisite data from disparate sources and assemble them together into a coherent simulation scenario. This laborious and error prone process presents a significant impediment to the goal of rapidly generating multiple COAs required for an AE. To support an AE, there must be a way to automate the production of COAs.

Securboration, a small business, performed research to address these problems. Their solution is the product SGen, which can semi-automatically produce COAs. The technical challenge and approach was to develop a robust data model which ties mission planning tools (SDT and JavaCOG) and disparate data sources (Modernized Integrated Database (MIDB) and Air Operations Database (AODB)) directly to the simulation. This robust data model is in the form of a flexible effects based ontology that defines the data and their relationships needed for COA generation.

With SGen populated with the scenario specific information from the SDT, JavaCOG and the data sources, the user can rapidly generate COAs. Figure 4 illustrates SGen's COA creation graphical user interface (GUI). At the top left corner are the *mission* and *phases* that are imported from the SDT. Below that the user has access to the *blue squadrons* that are available in theater. From here the user can select a specific squadron and instantly know the type and amount of aircraft available, and the airbase the squadron is assigned. A user also has the ability to create, delete or modify an order for a selected squadron. Below the squadrons is a basic timeline for the phases and overall mission. The right side of the interface illustrates the details of the COA. The tabs, in the upper right corner, keep track of orders for squadrons and target lists. Below the tabs in the *target assignment* section, the user can assign specific aircraft to a target. It also provides the ability to assist in effects based planning, through the *SOSA* (systems of system analysis (COG)) drop down menu. A user selects a target objective from the drop down menu, and SGen provides a list of assets that could indirectly affect that target. This information was directly imported from the COG model that was created in JavaCOG. SGen allows the user to target an objective directly or indirectly with a few mouse clicks.

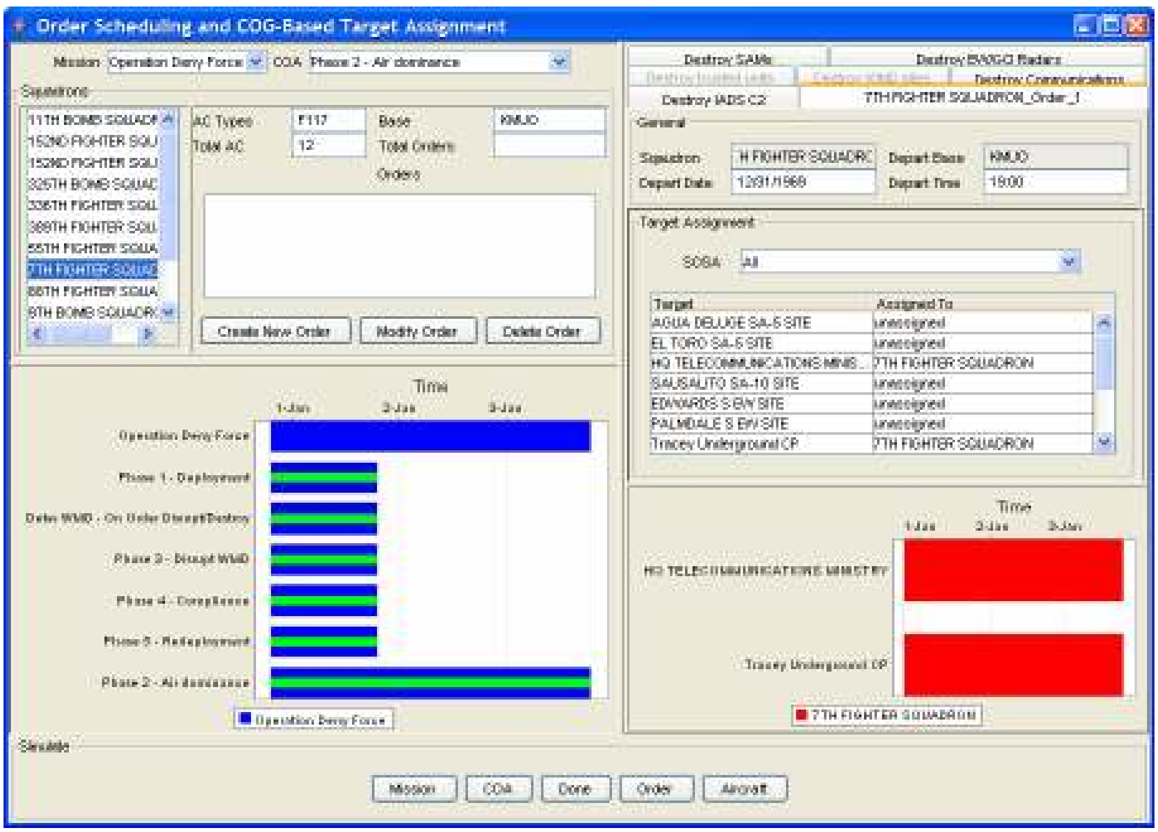


Figure 4. SGen COA Creation

After the user has created a COA, SGen produces the required files for the simulator. SGen currently supports two simulators; FSS and the Extended air defense simulation (EADSim) [4].

3.3 COA Analysis

Phase three in the JAEP is COA analysis. During COA analysis, each potential blue COA is wargamed against multiple red COAs. At a minimum there should be two red COAs: the adversary's "most likely" COA and the "most dangerous" COA. The main purpose of this phase is to identify the advantages and disadvantages for each blue COA.

The AE framework uses FSS to perform COA analysis. FSS is a flexible simulation test bed for development and integration of technologies that support dynamic COA/enemy course of action (eCOA) analysis. Some of the technologies developed are used in this AE framework, i.e. JavaCOG, SGen, and CASA which is discussed in section 3.4. Beyond the links to these technologies, FSS was chosen for its ability to simulate effects based operations. When planning EBO campaigns, military strategists analyze COG models to determine which targets will achieve higher level effects and overall military objectives. The interdependencies contained within the COG model are critical when utilizing a combination of direct, indirect, complex and cascading effects. To be able to simulate EBO, an abstract modeling methodology was developed to represent the COG models. This methodology is generic in nature and can be used to incorporate EBO concepts into virtually any event driven wargame simulator [5]. This methodology produced a generic EBO model that is capable of mimicking arbitrary EBO COGs. The modeling methodology was implemented within FSS to provide a capability to simulate EBO [6].

The COG modeling methodology provides the framework necessary for simulating EBO concepts. One of the vital EBO concepts implemented is the cascading event. This simulation event mimics the cascading nature of effects, which occur when a direct effect "ripples through an enemy target system, often influencing other target systems as well" [7], resulting in indirect effects or outcomes. In FSS, this occurs when one simulation object is influenced by another simulation object that it relies upon, e.g. if a factory is dependent on a power plant to operate, then an event that causes the power plant to be disabled will cascade to the factory causing the factory to be affected and possibly shutdown. A second essential EBO concept is the complex effect. This type of effect reflects the necessity of achieving cumulative effects on an object to have a desired effect. As described in [7], "cumulative effects result from the aggregate of many direct or indirect effects", e.g. the production capability of a factory could be interrupted by destroying or disrupting numerous transfer stations and generators, which are necessary to supply the power the factory requires to function. The ability to have simulation objects recover from both direct and indirect effects is also included in the EBO COG model, e.g. a factory can be affected by the loss of power caused by the destruction of a power line. The power line can be repaired (recover) at a given time, and cascades its recover event to the factory and any other simulation entity reliant upon this power line.

FSS's ability to provide simulation results that not only capture attrition, but the effects of those actions make it a powerful and unique simulation. For wargame simulations to be effective in an AE or effects based arena, they must allow users to evaluate multiple ways to accomplish the same goal with a combination of direct, indirect and cascading events (actions). This capability enables next generation Concepts of Operations regarding EBO to be assessed and evaluated within a simulation environment in much faster than real-time.

3.4 COA Comparison

In phase four, COA comparison, the wargaming results from phase three are compared against predetermined criteria [1]. As automated technologies are developed, an anticipatory environment could conceivably generate hundreds, if not thousands of potential blue and red COAs. The simulation of these COAs would result in a massive amount of data for analysis. Therefore, a technology is required to analyze and rank the wargamed results according to how well they achieve commander's intent.

The underlying technology related to the analysis of COAs is being pursued under the CASA effort. A key objective for CASA is to identify techniques to define appropriate measures of effectiveness (MOE) and measures of performance (MOP) for effects based COAs. Once these measures are defined, military analysts can use these metrics to rate and rank the relative merit of each COA evaluated through an operational-level wargaming simulation. The goal of the CASA effort is to identify a very low-level, fundamental, and common set of characteristics that, when aggregated, can be used to describe any MOE or MOP. Such an established set of characteristics would provide a direct means of comparison for disparate approaches that multiple COAs would unavoidably reflect.

The CASA approach uses a data representation ontology and schema to capture mission objectives, intent and COAs. The objectives, tasks, centers of gravity and measures needed to perform COA simulation analysis are extracted from the SDT and JavaCOG. These are captured in an XML instance document. The simulation results are parsed from simulation log files for relevant information (e.g. damage assessment, survivability, effectiveness) and also populated in the XML instance document. The results are captured with respect to the higher level COGs and displayed in a spreadsheet format using a tool called JavaRank. Visualization, when comparing many simulation results, is crucial for providing decision makers with insight into the strengths, weaknesses, and consequences of a particular COA. The JavaRank tool allows the user to quickly evaluate the top level composite COA scores as well as the capability to drill into each score for further analysis. This research is discussed more fully in reference [8].

4.0 Framework Integration

To develop the anticipatory environment framework, many dissimilar applications with little or no communication with one another had to be integrated. SGen can read outputs from the SDT, JavaCOG and databases. It parses these sources for the necessary information to populate the EBO ontology. After the user creates a COA from the GUI previously shown in Figure 4, SGen can automatically produce a parameter file of the blue COA. To simulate the COAs, FSS requires the blue and red COAs to be in the same parameter file. Furthermore, many events in FSS require exact matches between data in multiple parameter files. Therefore, a tool that would merge the two COAs and perform data verification testing was developed. After the COAs are merged, they are ready to be simulated within FSS. The simulation was executed and the results were logged. This log file needed to be parsed and new CASA files needed to be created to work in our framework.

Before parsing of the simulation log files could be accomplished, the CASA XML instance document must be produced. This file dictates the information that is required from the simulation log file. The parser uses the CASA instance document and JavaCOG to know what objects to consider and allows it to ignore objects that have no impact on the simulation and relating scores. An instance document was generated from the SDT file. This was done by parsing the XML file from the SDT and extracting the required information. From this file, the parse can acquire the high value objective targets the COAs are intended to affect.

All parsing work was done using the Java programming language, more specifically, regular expressions along with string functions. This required a parsing of the par files created by JavaCOG, hence the parsing tool made calls to JavaCOG to obtain a list of the appropriate objects to parse from the simulation. JavaCOG was used to determine the reason a given asset was targeted. JavaCOG accomplished this by reading in the objectives and the COAs. Given a target asset, JavaCOG traversed the dependency model by starting at the targeted asset and finishing when an objective node was found. JavaCOG makes the assumption that this asset was targeted to indirectly affect the objective node. This information is then passed back to the parser. Now the parser can begin to evaluate the log file.

After collecting all of the appropriate objects from JavaCOG, parsing was performed on the simulation files to create the results file which captured the events; such as persistence, survivability and degradation from all of the simulations. The simulation logs this information for specific objects in a text file. After multiple simulations, the parsing tool can parse the simulations in sequence and conclude with a finished results file. This file displays the results for each simulation run. The results file didn't display data for each particular object, but instead, separated the data by simulation number, displaying the averaged data for the simulation across the objects and their particular target set.

The instance and the results documents are then merged together to create the CASA populated instance document. In order to display the information in JavaRank, it must go through another transformation. The data concerning the objectives, COGs and simulation results are extracted from the populated CASA document and inserted into a JavaRank compatible file for comparison.

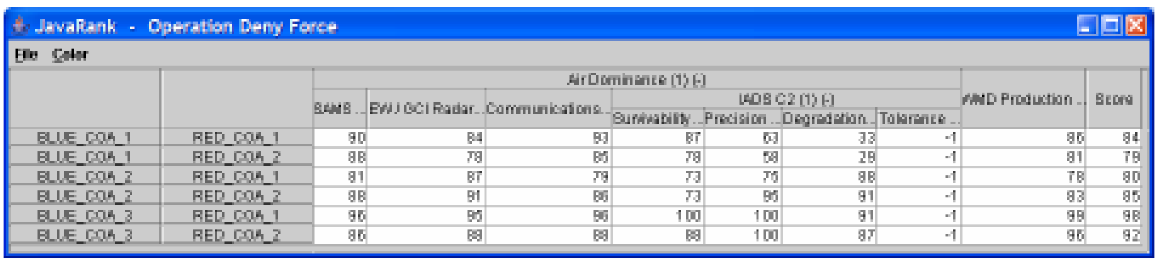
5.0 Demonstration

To demonstrate the AE framework and the integration of the tools and capabilities, a simple scenario was created; where the enemy forces are producing weapons of mass destruction (WMD) and planning on deploying them. To counteract the enemy, blue forces will attempt to stop the enemy's capabilities of deploying the WMDs. The AE framework described in this paper was used to analyze the mission, develop 3 blue and 2 red COAs, perform analysis of these competing COAs and analyze the results. This approach allowed us to demonstrate the concept of rapidly developing and assessing multiple blue COAs simulated against multiple red COAs.

The COG model, along with all of its dependencies, was authored within JavaCOG and contains 55 nodes. This model, which is significantly smaller than realistic models, was analyzed by humans and an initial high level blue COA was developed in the SDT. The goal of the blue COA is to stop the enemy's WMD capabilities. SGen was then used to rapidly create multiple instances (3) of the blue COA. The SGen produced COAs assigned blue assets to red targets within the COG model. The targeting approach of the three blue COAs was inherently different:

one was strictly attrition based and focused mainly on the red high value targets, and two focused on targeting the infrastructure with a goal of affecting the high value targets without destroying them. Two red COAs were created in the SDT and these focused on defending the red assets.

After creating our COAs, FSS was used to perform COA analysis to determine which COA would best achieve the goal of the campaign. FSS provides the capability to simulate the interaction of the blue vs. red COAs. Consequently, all three of the blue COAs were simulated against both of the red COAs, resulting in eight pairings. After the simulations were completed, the CASA process was used to measure and compare the results, which are shown in the JavaRank display in Figure 5. A quick look at Figure 5 shows that blue_COA_3 vs. red_COA_1 achieved the highest composite score. The user has the ability to drill further into the results to determine the nature of each score. This is currently a simple linear expression of the results. With further research, it may be more appropriate to assign weightings or some other computation method to the scores. However, the intent of this demonstration was not in the results of the analysis, but the speed with which the process of performing mission analysis through COA selection could be accomplished with the support of an anticipatory environment.



		Air Dominance (1) (-)							WMD Production ...	Score
		SAMS ...	EW / GCI Radar...	Communications...	IADS C2 (1) (-)					
					Survivability ...	Precision ...	Degradation...	Tolerance ...		
BLUE_COA_1	RED_COA_1	90	84	93	87	63	33	-1	86	84
BLUE_COA_1	RED_COA_2	88	78	85	78	58	29	-1	81	79
BLUE_COA_2	RED_COA_1	81	87	79	73	76	88	-1	78	80
BLUE_COA_2	RED_COA_2	88	91	86	73	95	91	-1	83	85
BLUE_COA_3	RED_COA_1	96	95	96	100	100	91	-1	99	98
BLUE_COA_3	RED_COA_2	86	89	89	89	100	87	-1	96	92

Figure 5. JavaRank Results

6.0 Future Work

To truly realize an AE there needs to be vast improvements in all of the tools described here. The AE framework presented in this paper was developed by integrating new technologies, most of which are in their infancy. While the integration of these technologies illustrated the concept of an anticipatory environment, they are not at technical readiness levels required for the warfighter. These tools require further research and development as well as time to mature, and there are additional technical obstacles to overcome to achieve our vision of an AE. Two major challenges reside in COG modeling and analysis, and the development of enemy courses of action (eCOAs).

Current COG modeling capabilities are static, brittle and very human intensive. There needs to be a capability to dynamically model and analyze COGs. This capability will not only improve the fidelity of the models, but also provides the capability to dynamically amend the model as the enemy changes and adapts. Additionally, the COG model used to demonstrate the AE framework was limited to 55 nodes. Operational COG models could conceivably contain thousands of nodes, and it is incomprehensible for humans to analyze and adjust the models in real-time. The reliance on COG models for EBO creates a critical requirement that the models are current and up to date. Therefore, a dynamic COG modeling and analysis capability are necessary to achieve the AE vision in support of EBO.

A major area for improvement in the AE framework is the generation of eCOAs. The framework currently uses a pre-scripted eCOA, which dramatically reduces the effectiveness of the COA analysis process. Since the eCOAs take significant time to develop, the planning process typically produces two; the "most likely" and "most dangerous". Furthermore, pre-scripted eCOAs are very brittle and are rarely relevant beyond the first several campaign events. To address this concern, research is being conducted on dynamic emergent adversary behavior modeling [9]. This will produce a red force that will act and react intelligently to blue actions based upon its own goals, intent and objectives. Also, the emergent behavior should be easily modified to rapidly produce different red actions. This will increase the number of eCOAs that each blue COA could be wargamed against, which will increase the robustness of the developed blue COAs.

An additional technology challenge that can not be overlooked relates to uncertainty modeling and management. The anticipatory environment described herein has significant reliance upon data and information, as well as modeling and simulation technology. Inherent in all of these is uncertainty. For decision makers to trust the results produced by an AE capability, it must provide an understanding of the level of risk or uncertainty associated with the results and recommended COAs. It must provide decision makers with an accurate perception of complex situations, their associated uncertainties and the risks associated with acting (or not acting) on information tagged with high-level uncertainty measures. This is a significant research challenge that must be addressed for an AE to be successfully implemented within an operational domain.

7.0 Conclusion

The process of performing mission analysis through COA comparison was expedited through the development and integration of the tools mentioned in this paper. While this process could conceivably take weeks or months, the framework allows this to be accomplished in days or possibly hours. This is a significant step towards the development of an anticipatory environment. However, significant research and development are necessary to fully realize an environment that supports decision makers and provides a capability to stay within the adversaries decision loop.

The creation of the AE framework provides a development environment for future research. The ability to integrate and test new tools/technologies within this environment helps to streamline research and development resulting in increased warfighter capabilities. The framework will continue to be expanded as new capabilities are created.

The AE framework presented strives to achieve the desired characteristics stated in the introduction. In the future this framework could provide the planning staffs with a better understanding of the mission space (past, present & future) and result in the generation of plan(s)/options that would "virtually checkmate" the adversary.

References

[1] Joint Publication 3-30, "Command and Control for Joint Air Operations", June 2003.

[2] Evans, Graham, Jones, Pioch, Prendergast and White, "Strategy Development for Effects-Based Planning", Military Operations Research Society (MORS) Workshop on Analyzing Effects Based Operations Read Ahead, McLean VA, January 2002.

[3] Gilmour, McKeever, Krause, Stirtzinger and Lehman, "Scenario Generation to Support Mission Planning", Proceedings of the 2006 Command & Control Research and Technology Symposium, San Diego CA, June 2006.

[4] http://www.eadsim.com/

[5] McKeever, Gilmour, and Hillman, "An Approach to Effects Based Modeling for Wargaming", Proceedings of SPIE: International Society for Optical Engineering Enabling Technologies for Simulation Science VIII: DSS 2004, Orlando FL, April 2004.

[6] McKeever, Walter, Gilmour, and Hanna, "Simulating Effects Based Operations", Proceedings of SPIE: International Society for Optical Engineering Enabling Technologies for Simulation Science IX: DSS 2005, Orlando FL, March 2005.

[7] US Joint Forces Command: Glossary, http://www.jfcom.mil/about/glossary.htm#C, March 2005.

[8] Hanna, Reaper, Cox and Walter, "Course of Action Simulation Analysis", Proceedings of the 10th International Command and Control Research and Technology Symposium, McClean, VA, June 2005.

[9] Lehman, Krause, Gilmour, Santos, "Intent Driven Adversarial Modeling", Proceedings of the 10th International Command and Control Research and Technology Symposium, McClean, VA, June 2005.