12th ICCRTS

"Adapting C2 to the 21st Century"

COURSE OF ACTION SCORING AND ANALYSIS

Topics: C2 Modeling and Simulation, C2 Analysis, C2 Architecture

Christopher Egan
Technical Point of Contact
Science Applications International Corporation
4031 Colonel Glenn Highway, Beavercreek, OH 45431
Phone: (937) 904-9251 (voice), (937) 431-2297(fax)
Christopher.Egan@WPAFB.AF.MIL

Jerry Reaper
Science Applications International Corporation
4031 Colonel Glenn Highway, Beavercreek, OH 45431
Phone: (937) 904-9150 (voice), (937) 431-2297(fax)
Jerome.Reaper@WPAFB.AF.MIL

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
2007		**00-00-2007 to 00-00-2007**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
Course of Action Scoring and Analysis	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) **Science Applications International Corporation,4031 Colonel Glenn Highway,Beavercreek,OH,45431**	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
Twelfth International Command and Control Research and Technology Symposium (12th ICCRTS), 19-21 June 2007, Newport, RI

14. ABSTRACT

The impacts of implementing effects-based operations (EBO) on course of action (COA) development and evaluation will be significant. Because EBO focuses on producing effects from military activities, as opposed to the direct result of attacking targets, there is an opportunity to develop a significantly higher number of COAs that achieve the desired effects. Consequently, EBO planning will significantly increase the number of evaluated COAs and the depth of evaluation. In order to evaluate these numerous COAs, which may achieve the same desired effects by substantially different methods, metrics must be found to adequately quantify their relative merits. Desired effects may be achieved though disparate COAs, such as propaganda campaigns versus major interdictions. The Course of Action Simulation Analysis (CASA) task was created to research metrics identification, data representation and scoring approaches. This paper introduces concepts behind CASA, chronicles task results to date, and finishes with a discussion of the scoring methodologies and capabilities developed during the CASA prototyping effort. Specific areas discussed include: mission-level simulations usage to examine multiple-hypothesis solutions; ontologies and extensible mark-up language (XML) metadata representations; COA metrics identification; development of tools for data reduction, comparison and visualization; and scoring approaches. Finally, lessons learned to date are discussed.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			
unclassified	**unclassified**	**unclassified**	**Same as Report (SAR)**	**49**	

ABSTRACT

The impacts of implementing effects-based operations (EBO) on course of action (COA) development and evaluation will be significant. Because EBO focuses on producing effects from military activities, as opposed to the direct result of attacking targets, there is an opportunity to develop a significantly higher number of COAs that achieve the desired effects. Consequently, EBO planning will significantly increase the number of evaluated COAs and the depth of evaluation. In order to evaluate these numerous COAs, which may achieve the same desired effects by substantially different methods, metrics must be found to adequately quantify their relative merits. Desired effects may be achieved though disparate COAs, such as propaganda campaigns versus major interdictions. The Course of Action Simulation Analysis (CASA) task was created to research metrics identification, data representation and scoring approaches. This paper introduces concepts behind CASA, chronicles task results to date, and finishes with a discussion of the scoring methodologies and capabilities developed during the CASA prototyping effort. Specific areas discussed include: mission-level simulations usage to examine multiple-hypothesis solutions; ontologies and extensible mark-up language (XML) metadata representations; COA metrics identification; development of tools for data reduction, comparison and visualization; and scoring approaches. Finally, lessons learned to date are discussed.

Table of contents

LIST OF FIGURES

1 ACKNOGLEDGEMENT

The research and prototyping accomplishments of the Course of Action Simulation and Analysis (CASA) task directly result from the vision, leadership and participation of scientists at the Air Force Research Laboratory Information Directorate (AFRL/IF) at Rome NY. The authors' efforts were in direct support of ongoing research at AFRL/IF into technology applications aiding military operators in accomplishing assigned warfighting efforts.

2 BACKGROUND AND INTRODUCTION

The military planning process depends upon analysis to anticipate and respond in real-time to a dynamically changing battlespace with counteractions. Complex technical challenges exist in automating these 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. Current generation wargaming technologies typically execute a prescripted sequence of events for an adversary, independent of the opposing force actions. A significant research challenge for wargaming is predicting and assessing how friendly actions result in adversary behavioral outcomes, and how those behavioral outcomes impact the adversary commander's decisions and future actions. The focus of this research was to develop technologies to assist decision makers in assessing friendly COAs against an operational-level adversarial environment. Utilizing high-performance computing (HPC) technology, it is possible to dynamically execute multiple simulations concurrently to evaluate COAs for critical elements related to execution and timing as well as overall effectiveness against a range of adversarial or enemy COAs (eCOA) [1].

Conventional wargames are also insufficient when it comes to evaluating modern campaign approaches. They focus on traditional attrition based force-on-force modeling, whereas modern campaign strategies employ and evaluate a mixture of kinetic and non-kinetic operations. The Air Force is pursuing EBO as one such modern campaign approach [2]. EBO focuses on producing effects from military activities, as opposed to the direct result of attacking targets. For wargames to be effective, they must allow users to evaluate multiple ways to accomplish the same goal with a combination of direct, indirect, complex, cumulative, and cascading effects. The overarching objective of this research activity has been to address the challenges of simulating EBO COAs in the presence of a dynamic adversarial environment, faster than real time. Such a system will allow planners to evaluate the effectiveness of today's alternative decisions and plans in tomorrow's battlefield.

Multiple research areas are under investigation: a simulation test bed; a scalable, flexible simulation framework; automated scenario generation techniques with dynamic update; intelligent adversarial behavior modeling; effects-based/attrition-based behavior modeling; and real-time analysis for comparing and grading the effectiveness of alternative simulations. The force structure simulation (FSS) test bed was developed in house to provide a capability to demonstrate the associated technologies necessary for performing parallel COA simulations faster than real time. The simulation framework will provide the foundation for rapid decision branch COA analysis [3]. Techniques to be able to evaluate multiple parallel COA simulations, as well as multiple branches, within a single COA were developed. Automated scenario generation techniques will enable the dynamic creation of simulation input files to support the concept of multiple parallel COA simulations [4]. Research on techniques to model adversarial behaviors will provide a simulation capability to anticipate potential adversarial actions for dynamic adversary COA analysis. A generic modeling methodology was developed in house to implement EBO concepts within virtually any modern wargame simulator and integrated within the testbed. The generic EBO model is capable of mimicking arbitrary EBO centers of gravity (COG), which contain dependencies and attributes of the target system. Techniques are also being investigated to define appropriate MOEs/MOPs for EBO COAs to help with the COA selection process.

2.1 WHAT IS A COA?

The definition of the term "Course of Action" varies significantly from person to person and application to application, as can be seen in the authoritative definition:

course of action — 1. Any sequence of activities that an individual or unit may follow. 2. A possible plan open to an individual or commander that would accomplish, or is related to the accomplishment of the mission. 3. The scheme adopted to accomplish a job or mission. 4. A line of conduct in an engagement. 5. A product of the Joint Operation Planning and Execution System concept development phase. Also called COA. (JP 1-02)

In general, the term expresses the concept of taking a series of actions to secure a desired set of outcomes. Militarily, "COA" is frequently used to describe actions at many different echelons, from theater level strategic activities through actions of specific units.

2.2 ROLE OF COA SCORING AND ANALYSIS

In current military planning, COA creation is performed by seasoned staff members with excellent backgrounds and knowledge of policy, capabilities, and the existing battlespace conditions. Their goal is to satisfy their commander's desires, as expressed in the Commander's Intent. This staff attempts to assess the most likely and most dangerous eCOAs and any specific dependencies that favor or damage COA success. The number of alternatives under consideration must be kept to a manageable number, frequently three. COA creation is generally based on the combined experiences of highly skilled and talented military personnel and proven rules-of-thumb. The results of this process are very often quite good and very frequently successful. However, existing staffing levels will not likely be able to continue these successes given simultaneous increases in both the number of COAs to be evaluated and the depth of evaluation. Rather, a means of augmenting and automating COA creation is required.

While there are multiple possible ways to automatically create COAs, all are sure to vastly increase the number of COAs available for a given scenario. This increase in the number of possible solutions must also be watched to prevent a corresponding increase in manpower due to analysis and selection of the generated COAs. The idea behind COA scoring is to try to automate the experience and knowledge present in military personnel doing planning today. This automation will then be applied to the generated COAs in order to prune the set of possible solutions to the best available. These solutions can then be analyzed by expert personnel for final COA selection.

The result of this effort is to analyze a vastly increased number of potential solutions, presenting the war-fighter with a choice between the very best.

2.3 EBO AND ITS RELATION TO COA SCORING

An emerging challenge within the command and control (C2) community is in placing EBO into action in practical terms. Key components of this challenge lay in developing processes, techniques, and tools for accurately predicting and assessing how kinetic and non-kinetic military actions and reactions impact the battlespace. These battlespace results, in turn, have consequences that spill over into the national and international communities. EBO is a structured C2 methodology that seeks to maximize positive results, minimize negative results, and balance the expected and potential costs associated with both. Figure 1 presents a view of the EBO concepts salient to this report. The impacts of practicing EBO on COA development and evaluation will be significant. Successful EBO planning has the need to significantly expand the tradespace of available COA options, increasing the raw number of COAs that must be evaluated as well as the overall complexity and depth of those evaluations.

Of particular interest to EBO are those areas that directly and indirectly govern and limit adversary commanders' decisions and future military actions. An initial step is to understand both the problem and solution processes sufficiently to develop organized approaches for implementing EBO-based planning. Once this understanding is in place, technologists can begin conceiving and prototyping automated tools that capture this knowledge. Such tools can act as force multipliers, enabling planning staff and decision makers to assess the numerous required EBO-based friendly COAs against similar numbers of potential adversarial actions and reactions.

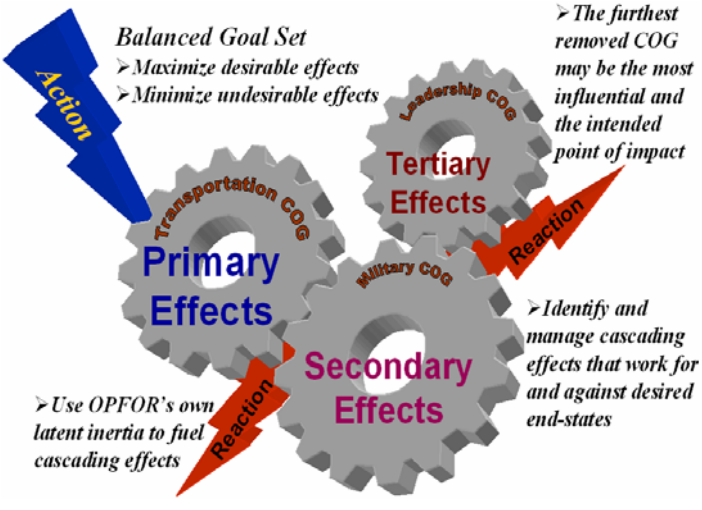
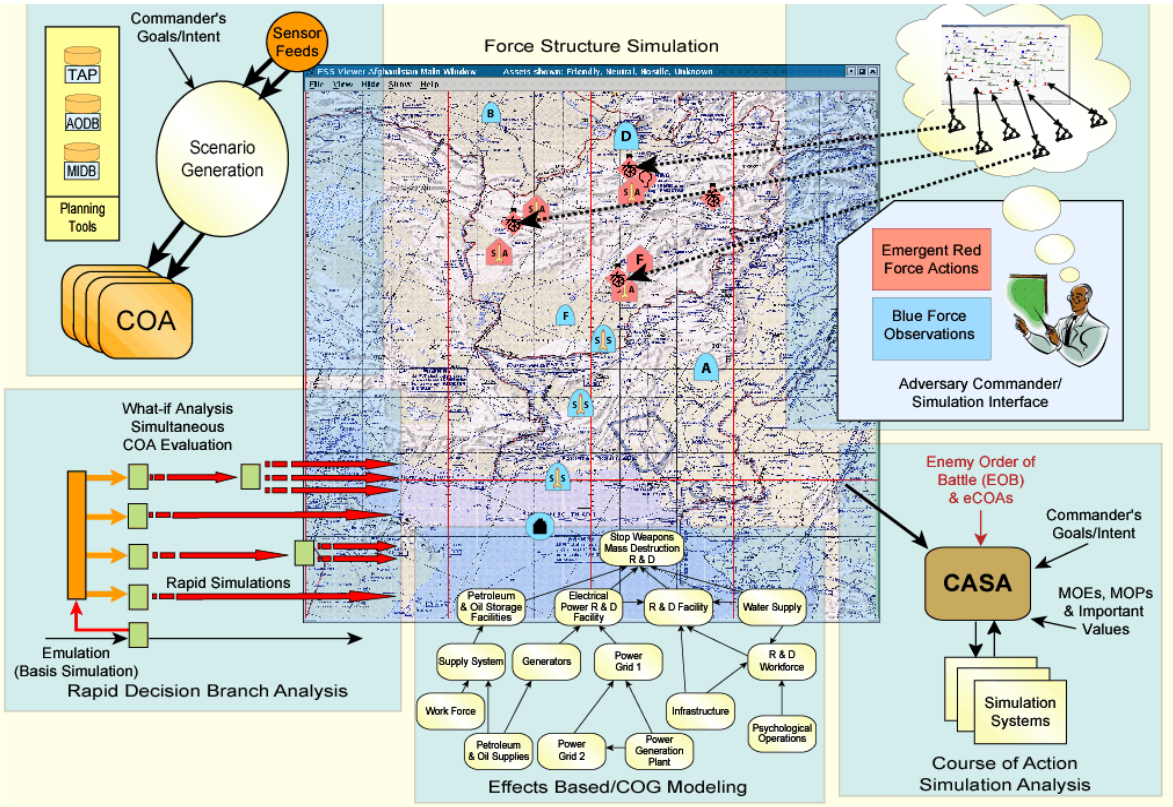


Figure 1. Key EBO Concepts

Conventional modeling and simulation (M&S) systems have largely been insufficient in evaluating EBO cascading effects within modern campaign contexts. Generally, such systems focus on traditional attrition-based, force-on-

force modeling whereas emerging and future campaign strategies employ and evaluate a mixture of kinetic and nonkinetic operations to reach desired end states. The Air Force is actively pursuing EBO-capable campaign M&S approaches and wargaming technologies [2]. Within these technologies, analysts exploit HPC systems to model and evaluate multiple hypotheses (i.e., COAs) for accomplishing the same goal with differing combinations of direct, indirect, complex, cumulative, and cascading effects. HPC concurrent simulation execution allows substantially faster-than-real-time simulation results, providing a practical means of determining the outcomes of many friendly COAs against a range of adversarial or eCOAs [1]. Each simulation is capable of yielding metrics on critical COA execution and timing elements along with overall metrics on friendly COA effectiveness. Additionally, the proposed HPC simulation system provides sufficient intervisibility within the model set to allow detailed examination of cumulative and cascading effects.

With a potential technological means to deeply examine EBO aspects of COAs via M&S, other related areas become progressively more important. Metrics must be established that adequately describe *and quantify* the relative merits of such disparate COAs as a propaganda campaign against a hostile populous versus a major kinetic military interdiction against an aggressor nation's military. The United States Air Force Research Laboratory Information Directorate (AFRL/IF) is leading several efforts to prototype a real-time decision support M&S-based toolset and methodology that allows rapid and thorough exploration of this expanding trade space. Figure 2 presents an overview of the integrated technologies under investigation.



? Figure 2. Real-Time Decision Support Architecture

3 TECHNICAL ACTIVITIES

3.1 SCORING STRATEGIES RESEARCH

Fundamentally, comparing differing COAs requires a quantification of the "goodness" of each element to be compared and some common basis over which the comparison is valid. These subcomponents must then be tallied to provide a composite score representing the "whole." Given the perceived disparate nature of dropping bombs versus dropping pamphlets, common basis can be quite difficult to find. That said, and assuming a common set of lower-level metrics can be defined, the simple production of a "score," (the roll-up of evaluations for a COA from bottom-to-top) is equally elusive.

In scoring a COA, human comprehension and approval of the scoring process is always necessary. This is true even when the COA is automatically scored, since the results of any automated method would require verification *by a person* before being approved to operate without oversight. This fact dictates that the production of a score must not only be correct, it must make sense to a human evaluator. This in turn, makes the implicit assumption that a human who is viewing the score can, at least over time, form some intuitive sense of what a score means, and what scores at one level imply about the scores at lower (hidden) levels. Consequently, the CASA approach to scoring had the goal of remaining as straightforward as possible, using a symmetrical approach to score roll-up so that humans can readily understand and readily develop an intuitive feel for the process.

In addition to being human friendly, the CASA scoring approach also needed to avoid known pitfalls in score assignments. Consequently, research into how humans optimally assign scores was performed. In an area known as "voting theory," options for differing choices (actually candidates, as voting theory relates to political voting) were evaluated to determine a format where people felt most pleased with the results. This research suggested that quantification of preference for each choice to a value in some arbitrary valuation produced conflicts when voters reviewed their selections. Alternately, when scores were forced to tally to set total (e.g., 100%), voters could intuitively create the relative rankings to captured preferences and were later more satisfied that those rankings were correct.

Several differing approaches were considered and compared using these criteria. Additionally, several approaches to possible low-level common metrics were considered.

3.1.1 Bayesian Networks

Mathematically, a Bayesian network (BN) is a probabilistic graphical model representing a set of variables with a joint probability distribution and defined dependence relations. In graphical terms, a BN is defined as a directed acyclic graph. Directed cycles are forbidden and nodes may represent any type of variable. Joint probability distributions are ones where the probability of one variable can directly affect the probability of another. A dependency relationship between the variables exists when this occurs. BN make use of conditional probability, where the probability of one event is conditional on the probability of a previous event or set of events. However, this is both a benefit and a limitation because all prior knowledge used must be applicable and trustworthy, or the "reasoning" results may be incorrect.

Practically, a BN is a probabilistic model of an environment or situation. Within a BN, the probability of an event occurring is directly related to the probability of a previous set of events occurring. This has a certain appeal to scientists and economists, and artificial intelligence research has used BNs to some degree of success. Within a decision-making system, actions and the relevant information leading up to actions would be inserted into nodes. These nodes are tied together by directed lines showing dependencies. All dependencies are quantified using equations that represent the dependent relationships.

While the use of BNs as a scoring approach for CASA was initially attractive, several basic issues arose in their usage. BNs are noted in literature (Niedermayer) for a distinct sensitivity to input dependencies. Differing subject matter experts (SME) often disagree on specific BN representations and consequently a common representation cannot be achieved. Additionally, the dependencies are captured in complex relationships between nodes. While this allows a rich set of tools for laboratory investigations, capturing the relationships of a small COA of several thousands nodes would require the specification of many times that in relationships. Having a human then develop an intuition about how scoring roll-up is unlikely and would vary between BN representations. Consequently, further BN research for the initial CASA implementation was abandoned in favor of identifying a simpler initial approach.

3.1.2 Attrition Based Scoring

Attrition-based scoring represents one approach to answering the need to identify a common set of scoring metrics that allow disparate COAs to be directly compared. The attrition-based scoring approach attempts to consider the kinetic effects of missions, both positive and negative. In researching this approach, several templates were constructed to account for how the results of kinetic actions affected numerous facets of the battle space, including but not limited to, adversary forces; civilian populations; economics; and political, religious, and cultural infrastructures. What quickly became obvious was that each examined application of kinetic force had numerous exceptions. When the templates were combined and revised to attempt to account for all variations, they became

very large and were generally sparsely populated and unwieldy. Their sparse nature forced abstraction to allow for direct comparison, with each abstraction specific to the COAs under examination. Additionally, attempting to allow for EBO considerations expanded both template size and complexity. Following numerous failed attempts to find a means to use this scoring approach, a more fundamentally abstract approach was researched.

3.1.3 Task and Effect-Based Scoring

The main emphasis in the task and effect-based scoring model is to account for the relative importance of the individual tasks on achieving the overall desired effects. The viewpoint here is that although a given COA may inflict dramatically more casualties than are sustained, it could still prove to be a poor COA. For example, suppose the purpose of a COA is to destroy the enemy's ability to utilize weapons of mass destruction (WMD). It achieves 25% success on all missions without losing a single asset, but the COA would not be considered a success. The enemy will still be able to utilize WMD even though they sustained more damage than they inflicted.

The other major aspect to this approach is the modeling of EBO and cascading effects. This is used to account for desired results that are not achieved through direct action. For example, if we decide to disable a factory by destroying the power plant supplying it with power, we must both account for the fact that bombing the power plant facilitates the desired goal and to ensure that success in bombing the power plant produces the expected results.

The basic components to this approach are to break the COA into logical tasks, identify scorable metrics that are common to all tasks, and ensure that the COA achieves the desired results. In order to break down the COA, we first utilized the Universal Joint Task List (UJTL, CJCSM 3500.04D, 1 August 2005) breakdown currently in use by military planners. This breakdown decomposes a desired end state for the scenario into manageable parts down to individual missions. From missions, it is further broken down into metrics that are common to all missions. These are metrics that were developed during the CASA program through working with SMEs in COA planning. In order to ensure that success of the individual missions resulted in achieving the desired effects, the concept of *influences* was developed. We will now examine this approach in more detail.

3.1.4 Metrics Breakdown

One of the chief problems discovered with the attrition-based approach was that the strategic decomposition of Commander's Intent into strategic goals and actions was largely unaddressed. Instead, the statement of intent and subsequent strategic planning was assumed to have occurred. The rationale behind such plans was lost, while actual plans were directly incorporated into the information template. Thus, much information of value to EBO was unavailable. Also missing were direct ties to the concept of COG at national and theater levels.

In order to better capture the strategic plan of a COA, as well as break that plan into logical pieces to facilitate scoring, a restructuring of the COA was needed. A high-level view of this restructuring is presented in Figure 3.

At the top most level of the CASA information hierarchy is the Commander's Intent, stated as a list of desired end-state conditions that must be present to successfully satisfy the commander's goals. These conditions may relate to international-, national-, and theater-based goals and will likely address areas such as economics, diplomacy, military capabilities and security.

The next level in the hierarchy relates the Strategic Effects required to transition from current conditions to those desired end-states expressed in the Commander's Intent. Negative effects (those moving away from the desired end-states) are also noted, recorded, and addressed. This level identifies the relevant Diplomatic, Information, Military, Economic (DIME) instruments, their roles, and the effects that they are required to produce. An expected focus area addressed by this level is how the military instrument will be used. The Strategic Effects level then further decomposes into the Operational Effects that support or hinder realizing each defined Strategic Effect. Similarly, the Operational Effects level decomposes into the Joint Operational Tasking required to achieve those Operational Effects. These tasks in turn decompose into Operational Missions required to successfully satisfy those tasks. An additional component of information at each level is the contribution toward success that each effect, task, and mission has on elements one level up. For example, within the information for each Operational Mission is the contribution that the mission makes toward achieving the Operational Task that it supports. This hierarchy provides a simple and intuitive structure of how a top-level goal decomposes into a set of detailed actions, as well as providing an equally intuitive means to quantify the contribution of each low-level action toward higher-level goals.

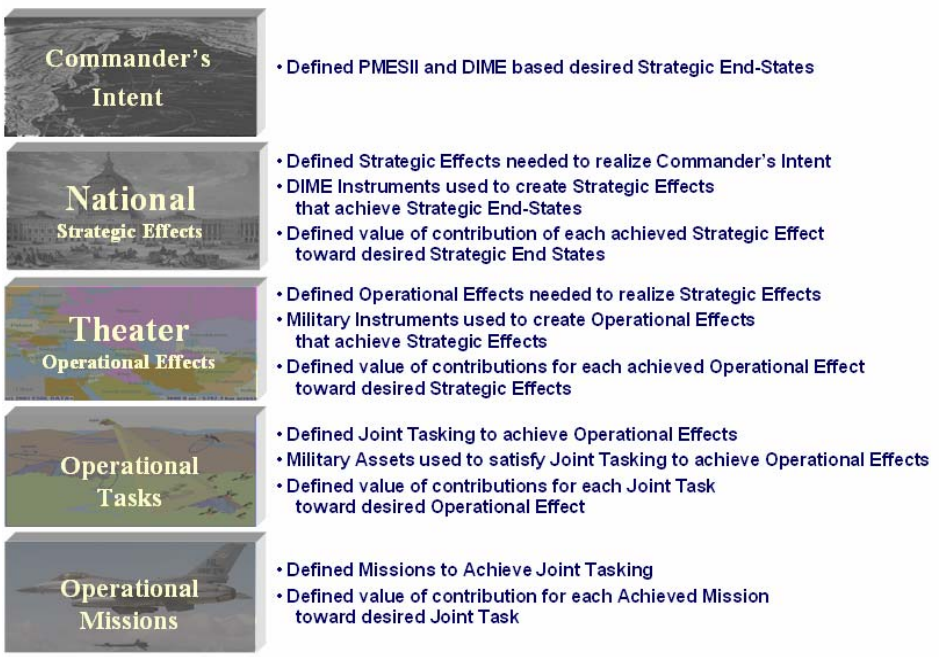


Figure 3. Notional CASA Information Layout

One of the main reasons that the UJTL breakdown was modeled in CASA was because it is already working in practice. This provided a ready-made breakdown that had already been proven effective and had evolved over the years to become better. Rather than inventing a separate breakdown that would have potentially been found to have flaws, the CASA scoring approach was able to capitalize off of the lessons learned and experience developed through years of use.

A second reason for adhering to the UJTL structure was in order to give the scoring approach increased familiarity with military planners. If an approach like the one implemented by CASA is ever fielded, users will be able to recognize many of the concepts in the scoring breakdown, decreasing the learning curve for the tool.

3.1.5 Development of Common Metrics

The breakdown of a Commander's Intent to the level of missions is not a fine enough granularity to automatically generate scores. Common metrics that would be applicable to all types of missions were identified by talking to SMEs with field experience. It was determined that any type of mission could be scored on three measures of merit (MOM): Mission Effectiveness, Mission Efficiency, and Timeliness.

The common metrics (MOM) were further broken down to map to actual pieces of data. MOMs consist of MOEs, which are broken down further into MOPs. This is done to isolate related events and score them accordingly.

As shown in Figure 4, each type of MOM has its own subset of MOEs. Mission Effectiveness MOMs consist of measures that represent every possible type of conflict. Mission Efficiency MOMs include the MOEs that measure expended resources, and Timeliness MOMs break down into MOEs that measure time-over-target (TOT) and landing events based on expected values. While the figure represents the current configuration, measures can be added or deleted as needed.

Just like MOMs, MOEs break down into their own subsets of MOPs. These MOPs take in event information from the missions and score them. Engagement event information is used in Disrupt, Destroy, Disable (DDD) MOPs and TOT MOPs. Land event times are used for the Landing Time MOPs and the Start mission events use its time value for TOT MOPs. This allows a clear division to be made of event information gathered from the scenario. Each measure has a value and a weight. The value is the score for that particular measure and the weight is that measure's degree of influence in making up its parent's score. In CASA, the Effectiveness MOMs carry the most weight in mission scoring at 70%, followed by the Timeliness MOMs at 25%. Values of measures are normalized to a range of 0 to 1 so that they cannot exert more influence on their parent measure than is specified by their weight.

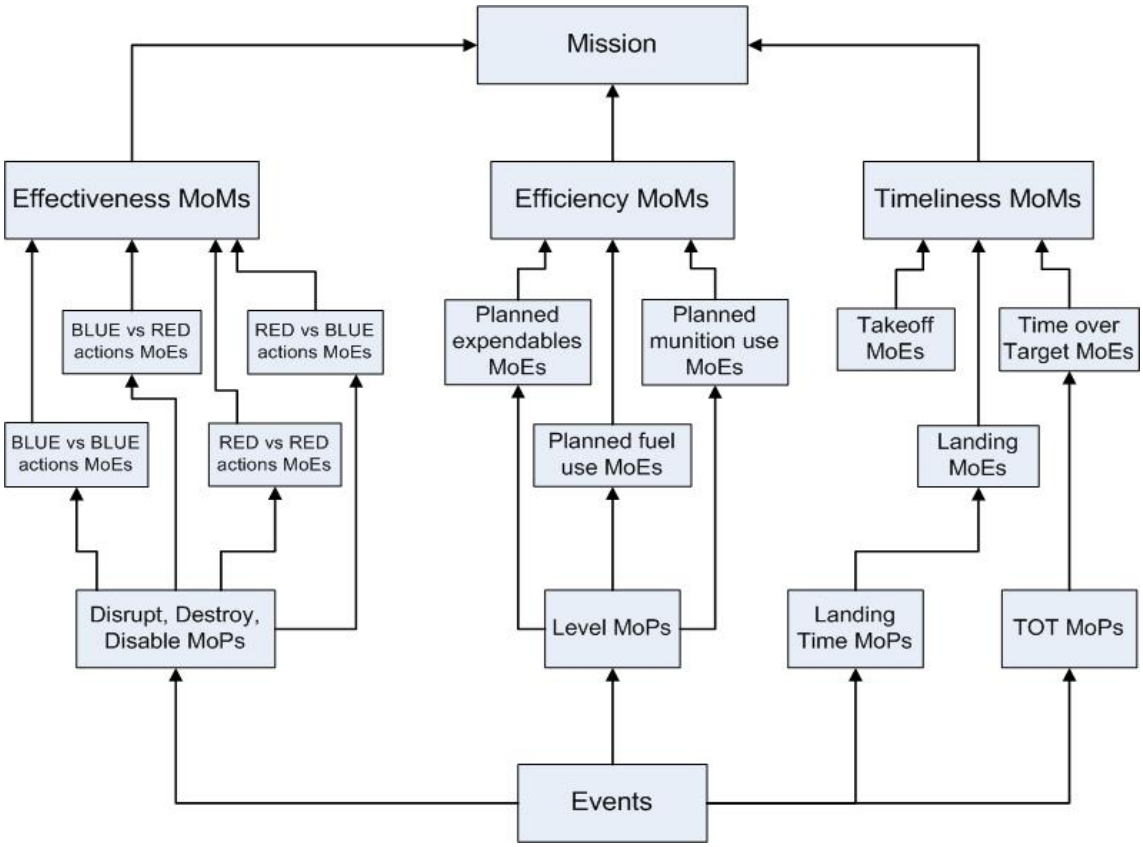


Figure 4. Mission MOMs

3.1.6 Influences

Influences are a separate scoring mechanism from the metrics used to capture score-on concepts and EBO effects that may not map directly to data from the simulation. They are used to bias the score either positively or negatively based on whether a set of conditions is met.

Influences were developed to solve a problem where the score of the overall COA was not reflected by the lower level scores. For example, an Operational Task to eliminate enemy radar would score well if the component missions all scored well even if the enemy radar actually remained operational. By allowing for an influence to adjust the score negatively if the true intent of the measure failed (e.g., radar still operational), there is a mechanism that allows the true intent of the measure to be scored.

This capability also allows for better scoring of COAs utilizing EBOs. By specifying an influence that heavily adjusts the metrics achieved through EBO effects, the score will reflect the success or failure of the intended effect. If we have a COA where a large amount of otherwise inconsequential actions are taken to produce a desired effect, we can add in an influence to make sure the desired effect was achieved. In this way, the achievement of the desired result will be a main driver of the score instead of the results of the inconsequential actions.

Influences are primarily comprised of three things: the measure to adjust, the conditions that will cause them to be applied, and the adjustment factor they have on the score. Influences can be specified at any level in the score breakdown. Any measure that is scoring an abstract concept could utilize an influence to account for the success or failure of that concept.

The set of conditions that cause an influence to be applied can conceptually be any measurable condition. In order to facilitate automation, the focus in developing influences has been to create conditions that are not subject to interpretation. With this in mind, the bulk of the research done was on conditions that evaluate the state of assets or COGs, such as a specific asset is disabled for a given time or that a COG is operational. In this way, influences

provide a key mechanism needed for ensuring intended results are achieved and a means to build in sanity checks to the score.

Once an influence's conditions have been evaluated and are found to be true, that influence's adjustment to the score must then be applied. This adjustment could be any type of equation specified by the user. For the research done so far the adjustment was limited to a multiplier on the score. The commutative nature of multiplication removed issues dealing with ordering the application of influences. For example, if an Operational Task has three influences specified and the conditions of two of the influences are evaluated to be true then the order in which their adjustments were applied would need to be specified if their operations were not commutative. The simplistic nature of multiplication also allowed for an easier understanding of the effects of influences at various levels of the score. While further research would need to be done in order to figure out the optimal combination of power and understandability, multiplication provided a good balance of each for the research done to date.

3.1.7 Scoring the COA

Scoring a COA is broken down into two basic parts: scoring the lowest level MOPs and scoring all of the higher-level measures. Scoring MOPs is different from scoring any of the other measures because they are the leaf nodes in the score breakdown tree. As such, they are not composed of other measures but instead are scored based on data taken directly from the simulation results. All of the high-level MOPs are scored based on how well their constituent MOPs scored. In this way, MOPs are building blocks used to build up higher-level scores.

The score of a MOP is determined using a formula based on how well the actual value achieved the intended value. This formula could be any equation and could be different for the various types of MOPs used to measure various aspects of simulation results. The result of the scoring for each MOP is that it will have a value calculated for how well it was achieved.

Once all of the MOPs have been scored the first step in the score roll-up process can begin. This is an iterative step where all of the measures at a given level are scored prior to calculating the next higher level. The next level above MOPs is the MOEs, so their base scores would be calculated by summing the weighted total of the MOPs that make up each MOE.

After the base score for each MOE is calculated from the weighted-average total of the MOPs, any specified influences are evaluated for their modification on the score. Any influences whose conditions evaluate to true then have their adjustment applied to the base score of the measure. After the influences have been applied, the final score for the MOE is calculated. This roll-up process is then repeated up each level of breakdown until the Commander's Intent is scored.

3.1.8 Role of Cost in the Score

In addition to the score of how well a COA achieved its intended results, the cost of implementing that COA is also calculated and presented to the user. There may be certain COAs where achievement of desired results is paramount regardless of cost, others where significant differences outweigh small differences in results achievement. This metric is presented as a peer to the COA score in order to serve as a separate decision point for the user.

The cost could be calculated as a component of the COA score, but this would hide the information and dilute the score of how well the desired results were achieved. By presenting cost at the same level as the score, the user has the information necessary to evaluate COAs first on success and second on cost. This prevents a mediocre COA with very little implementation cost from being artificially inflated. Similarly, a COA with widely divergent results but a stable cost will not be artificially smoothed by the inclusion of cost in the score.

3.1.9 Information Storage and Display

The original work for CASA centered on a spreadsheet approach for data storage and representation. However this approach quickly became exceedingly complex for the data and relationships of even small COAs. In an effort to represent data beyond two dimensions, ontologies were evaluated. An ontology is a relational model of data. Instances are created and grouped into classes based on their attributes. Inheritance-based specification of these classes ensure uniformity and data independence in the constructs. These logical groupings of data into higher-level concepts provide a clear correlation of data into information.

The tool chosen was Protégé, developed by Stanford University. Protégé is a free, open-source utility created to model ontologies. Protégé is Java-ready, and basic functions to communicate with the ontology were accessible via an application programming interface (API). The API allowed for ease of data population and extraction, facilitating automation. Plug-in extensions are also available for additional functionality.

Easy navigation helps in understanding of overall flow and conveys the big picture. By allowing the user to navigate between different levels and related elements, they are better able to understand the relations between the different objects in the scored ontology.

It was determined through the course of prototyping different scoring solutions that much of the goals for information representation can be achieved through proper use of the graphical user interface (GUI). Ontologies proved to be particularly useful for presenting logically grouped information to users. The inheritance aspect also facilitated presenting a hierarchy to the user so they could focus on only the desired level at one time. Navigation between related objects was also very easy through the ontology. Each sub object could be viewed in detail by simply clicking on it. While better GUIs may be found that would present information to the user in a still more intuitive fashion, ontologies were found to be very useful during the research and prototyping effort.

3.2 DEVELOPMENT OF PROTOTYPE

The main focus of the prototyping effort was to automate the task/effect-based scoring approach. The intent of this effort was to more fully demonstrate the capability to automatically generate a score for a COA as well as to refine the scoring process.

Automation was a key step in refining the COA scoring process because it allows for greatly increased experimentation and evaluation of different COAs. Whereas in previous years, the scoring methodology was only able to be tested against a few COAs to test its validity, automatic scoring enables a greatly increased number of COAs to be examined and the results of the scoring process tested against a larger set. Automation of the scoring process also allows for changes to be easily tested and propagated through the system. In this manner, refinements to the assumptions scoring logic can easily be made and their merit verified or refuted without time-consuming manual propagation of the changes. The mechanization of the scoring also adds an increased degree of accuracy to the score, ensuring that all rules are applied uniformly throughout the process without the need for rigorous manual verification.

Perhaps one of the biggest benefits of the automation sought in the prototype effort was verification of the idea that a score could be automatically calculated by putting the theories developed in prior years into practice. This rubber-meets-the-road milestone was intended to clearly demonstrate that ideas that worked on paper could actually work in practice.

As an overview, the goal of the prototype was to be able to take a template ontology defining basic data structures and relationships and populate it with the actual assets and events from a simulation run. After the ontology was populated, it would be scored based on the task and effect based strategy developed against specified scoring criteria. The end result would be to produce a populated and scored ontology that could be analyzed by a user. Figure 5 shows the flow of events for scoring a COA with the prototype. The area enclosed by the dotted red line represents components developed by the CASA prototype.

3.2.1 Ontology Creation

The development of the Year III prototype focused on development of an ontology for storing and viewing information, reading simulation files for data population, specification of scoring criteria and score automation. Each of these activities will now be described in more detail.

The use of the ontology tool, Protégé, served as both the back-end data storage as well as the front-end user interface for the Year III prototype. Constructing an ontology in Protégé is analogous in many ways to constructing a database architecture. The same process of defining the types of information to be stored, organization of the pieces of data and relationships need to be performed in both technologies. One of the benefits of the Protégé tool was that after completing the data definition step, a functional front-end is also available for viewing and entering data in the ontology.

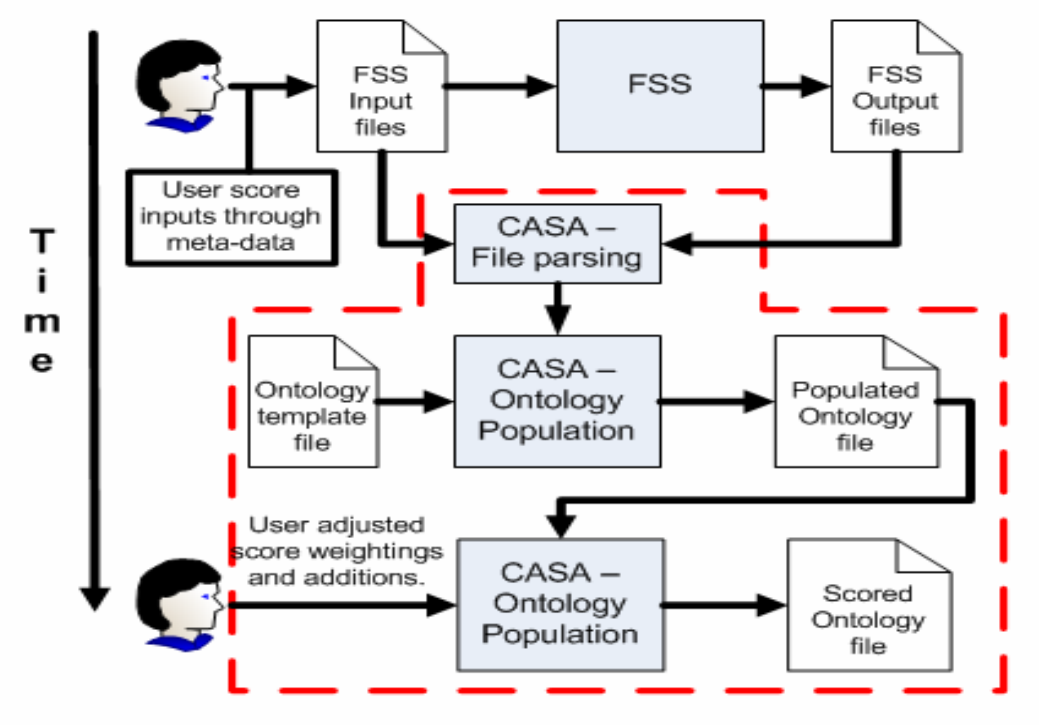


Figure 5. Year III Prototype Flow of Events

The use of the Protégé tool's basic front-end allowed the main focus of the Year III effort to concentrate on development of the scoring algorithms as opposed to having to spend time implementing a custom front-end. While this approach did limit the control that could be exerted over the user interface, the provided interface was sufficient for the prototype effort.

The process of actually constructing the ontology was done by modeling the data that would be read from the simulation (i.e., what types of assets are there, what attributes do they have, and what kinds of things can happen to them), modeling the scoring structures (i.e., all of the different measures and their relationships) and determining default data values.

The support of inheritance greatly facilitated the modeling of simulation data, scoring structures, and their respective relationships. This allowed basic concepts to be modeled at a high level and then subclassed to model more specific objects. By modeling data in this fashion, the constructs could be accessed in a polymorphic fashion, greatly enhancing the robustness of the prototype.

Although efforts were made to construct the ontology such that its structures could be compatible with any simulation, there are probably areas where the modeling closely matches how assets are modeled in FSS. This occurred both because FSS was the only reference used to develop the prototype and the benefits of developing of an independent model could not be addressed within the Year III effort cost and schedule.

Default values are specified by the template ontology. Upon initial creation, every measure receives the default desired value for that type of measure. This value can then be overridden by the user through one of the other mechanisms if desired. Figure 6 shows the default weight specified by the template ontology for MOMs. All created instances of MOMs will initially have these values as their weight. As will be discussed in the assumptions section below, defaults are useful for generation of a score without the user needing to specify much data. They also ensure that all instances are fully populated with the values necessary for scoring in case the user omits specifying some values.

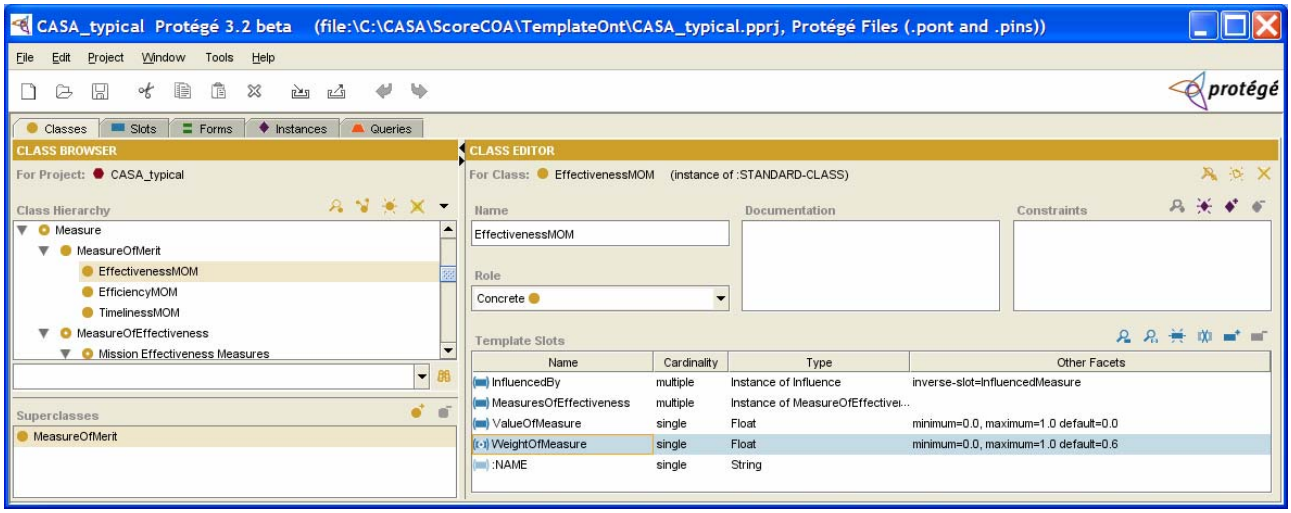


Figure 6. MoM Default Weight as Specified by the Template Ontology

3.2.2 Meta-Data

After the ontology was created and populated, the next step was to provide a means of specifying the specific scoring criteria for the COA to be scored. This mechanism needed to be outside of the template ontology so that a template could be used to model default values for multiple COAs. The method also needed to allow for storage of the criteria so that extensive user action was not needed in order to score a COA. With these goals in mind, the approved approach was to utilize a file of meta-data to specify scoring criteria.

The meta-data specifies values for actions to take place in the simulation, the UJTL breakdown and specification of influences. The UJTL breakdown is specified because FSS is a mission-level simulation, so the higher-level breakdowns need to be added prior to score generation. The values that are read from the file are then applied to the specific measures and influences that they reference in order to override the default values. This mechanism is useful for saving specific adjustments a user wants to make to the scoring of a COA and to be able to repeat them. The repeatable aspect of the meta-data enables different runs of the scoring process to be performed, allowing for results from different seed values to be evaluated as well as simply being able to recreate the same scored ontology from the input files.

3.2.3 Scoring Assumptions

One of the key learning points derived from creating a prototype to automate COA scoring was the discovery of how many assumptions a human scorer naturally makes. The process of automating the scoring process through code made this abundantly clear. Since computers cannot make any assumptions, every decision point that was taken for granted during human score development was analyzed for validity. This forced review of assumptions was an unanticipated benefit of automating the scoring process that resulted in improvements to several areas of the scoring algorithm.

The main reason why assumptions are needed is that they reduce the amount of information that the user needs to input into the system in order to generate a score. For every assumption that is built into the toolset there is a corresponding piece of information that the user no longer needs to enter unless they want to override the default. For example, rather than forcing the user to say that destruction of opposing force military assets is a good thing every time an asset is destroyed, we can build that assumption into the toolset. In this way the user now would only need to enter information into the scoring process if that default case were not true (i.e., destruction of an enemy military asset was a bad thing). As long as the assumption is true more often than it is not, we have reduced the overall data entry into the system.

Although every assumption corresponds to a piece of information that no longer needs to be entered, it also corresponds to a rule placed on the system. For this reason the number of assumptions was kept to the minimum necessary to generate a score without encumbering the user. The concern being that the more rules placed on the system, the more users would accept the default cases without evaluating whether or not they made sense for the

specific situation being scored. For this reason, assumptions were only added that were believed to be true for most cases as opposed to just the majority of cases. Forcing the user to enter values for cases that often go either way prevents developing a mindset of accepting the default.

In order to prevent the default values from becoming a constraint on the system, efforts were made to expose the assumptions made through the GUI. This was done to give the users more control over the system and to change the way that the default settings work. This allows for basic concepts that are true for U.S. military strategy to be reversed in order to model COAs that might appeal to terrorists or other asymmetrical warfare combatants. This control was made available to the users by making the scoring software very generalized and putting default values in the template ontology. These default values are then inherited by every instance created in the populated ontology. By putting the information in the ontology, a finer degree of control is also achieved. Instead of specifying traits for a group of assets, that group can inherit a default trait and individual assets can be overridden with specific values as necessary.

An example of the assumptions built into the ontology would be that the default classification of power plants is to consider them as civilian targetable assets. In this way, power plants that are intended to be attacked and are attacked successfully will positively influence the score. A specific instance of a power plant could be deemed to be classified as civilian nontargetable (perhaps because it only powers a hospital) in which case damaging it would negatively influence the score.

3.2.4 Entry of Desired Values and Scoring Criteria

There are three ways to enter desired values, influences, and other scoring criteria into the CASA software toolset. These methods are default values specified in the template ontology, user-defined meta-data, and user-entered information through the GUI. These sources of scoring criteria are applied like a series of lenses in order to achieve the right score.

The first lens is the default values from the template ontology. These values provide the coarse score adjustment criteria, defining general concepts of what actions are positive and negative as well as the relative weights of broad categories of actions. The default values also ensure that all measures are fully populated so that they can be scored. By providing an initial value for all scoring criteria, the user is relieved from ensuring that all necessary values are defined through some combination of the three entry mechanisms.

The second lens is the user-defined meta-data. The values provided through meta-data are specific to the individual scenario being analyzed, as opposed to the default values that could be applied to many scenarios. In level of granularity, the default values are analogous to a far-sighted or near-sighted lens whereas the meta-data is the specific prescription needed by an individual.

The final lens is user inputs through the GUI. This method is most effective for testing a hypothesis on the score. Once the score has been calculated using the default values and meta-data provided, the user is then able to adjust to the scoring criteria through simple GUI input of data. The score can then be quickly rerun with the user changes for analysis of the impact of the changes. By allowing the user to rapidly test the impact of changes, the GUI input facilitates robustness testing and iteratively improving the score criteria specified. Typically, once an entered criterion is found that has a desired effect on the score, it will be either added to the meta-data or changed to the default value in the template ontology so that it becomes saved and can be applied to future scores. Continuing with the lenses analogy, this lens is typically used to try slightly lower or higher values; once the correct value is identified, it would be moved to the meta-data.

3.2.5 Core Scoring Software

With the specifics of the ontology and simulation encapsulated by the interface classes, the core scoring software was able to focus solely on automating the scoring techniques pioneered in Year II. This process was broken down into three main tasks in the code: scoring MOPs, evaluation of influence conditions, and score roll-up.

3.2.5.1 Scoring MOPS

Scoring MOPs was a major focus of automation because they are the lowest level measure and as such are the only measure whose score is not based off its children's measures. While several different types of MOP were identified,

all are scored in the same manner. The basic types of MOPs created in the Year III prototype are Time MOPs, Duration MOPs, Level MOPs, and DDD MOPs.

Time MOPs are used to measure how close to a desired time an event occurred. This MOP is typically used to measure how well an asset achieved its desired TOT, as well as how well it achieved on-time landing.

Duration MOPs are used to measure how close an event duration met its desired duration window. For example, this type of MOP could be used to score how well a mission to disable a command post for a four hour period achieved that disabling for the desired period.

Level MOPs measure how well an actual level achieved the specified desired level. This could be used to score how well the actual fuel usage for a mission compared to the planned fuel expenditure.

The final type of MOP implemented for the prototype was the DDD MOP. This MOP is actually a specialization of the Level MOP. As such, it measures how well a target was disrupted, disabled, or destroyed to the desired value. This specialization was created in order to allow different default values to be created for these types of MOPs as well as to separate them from other Level MOPs both in the code and in their presentation to the user.

Because MOPs are at the end of the chain, they derive their score from how well actual result values taken from the simulation achieve desired results. The desired values for MOPs come from a combination of the three sources: default values, meta-data, and user inputs through the GUI. The actual values are populated through the data parsing and correlation done by the simulation interface. Once both the desired and actual values are populated, the score for the MOP is calculated.

The equation to score the MOPs is composed of two parts. Every MOP has a desired high and low value as well as an actual high and low value. First, the actual low value is scored against the desired range, and then the actual high value is scored in the same manner. After both scores are calculated, they are averaged together in order to produce the overall score. For MOPs with only one actual value (e.g., landing time), the actual time is used for both the actual low and actual high values. This leads to a little extra processing overhead for MOPs with only a single actual value but allows for uniform processing of scores.

In order to generate a score for each actual value, the following formula is used:

1 - ((amount outside of desired threshold)/desired range)

This formula returns a value of 1.0 for actual values that fall within the desired range. The score reduction for achieving an actual value outside of the desired range is dependent upon both the amount outside of the range and size of the range itself. Because of this, a MOP that is off by 5 with a desired range of 20 will score higher than a MOP that is off by 5 with a desired range of 10. The rationale behind this decision is that if a small range is defined then there is a lesser tolerance for error than if a larger range was defined. This approach also necessitates less information from the user in order to generate a score. Instead of specifying both a desired range and an equation relating how the score should react if the desired range is unmet, just a desired range is needed. The reduction of user input was a key goal in order to facilitate automatic score generation.

While the MOP scoring approach implemented proved valid for most of the cases developed, it would most likely need to be expanded in a fully developed system to allow for more user control. The ability for a user to override the default score behavior with a user-defined equation seems like a useful feature, but was not developed as part of the Year III prototype due to time and tool constraints.

Although there are different types of MOPs to measure different metrics, all of the MOPs are scored by the same algorithm. This was done both to simplify the automation as well as to handle the scores in a uniform manner. While the same scoring equation was applicable to all of the MOPs developed in Year III, it would not necessarily be true that MOPs developed in the future would be able to be scored in the same manner.

3.2.5.2 Influence Conditions

The second major development area of the core scoring software is code to evaluate whether or not influence conditions were met. This must be done in order to determine whether each influence's modification on the score is to be applied. If all of the conditions for a given influence are met, then that influence will be applied to its measure.

The influence conditions implemented for the Year III prototype operate as checks that specific assets are either in or not in a set of predefined states for a specified time period. An example influence condition might be that a

command post was in the disabled state for all of day three of the scenario. If the command post becomes disabled during day two and remains so until day four, then this condition would be met. Another example would be a condition to check that an air defense site was operational at a certain time.

The set of states for which to check assets revolves around the DDD paradigm, with operational as a fourth state to check. Figure 9 shows the influence conditions that were implemented for the prototype. The conditions with multiple states will evaluate to true if the specified asset is in any of the states. Not all possible combinations of the states are implemented because some combinations were considered unnecessary. For example, there is no "Destroyed/Disrupted" condition because destroyed/disabled/disrupted can be thought of as doing progressively less permanent damage. It therefore seems unlikely that we would want a condition to ensure that an asset was in either end of the spectrum without allowing for the middle ground.

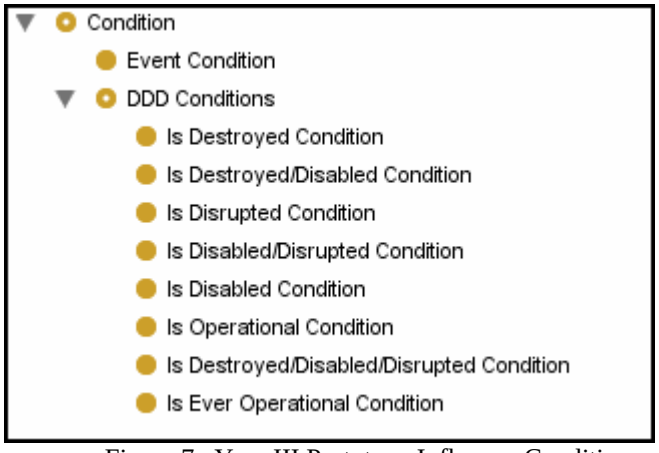


Figure 7. Year III Prototype Influence Conditions

3.2.5.3 Score Roll-Up

After the MOPs are scored and the influences are evaluated, the score roll-up process can begin. This process is merely the automation of the scoring procedure outlined in Section 2.2.1.3, "Task/Effect-Based Scoring." The measures are walked backwards up the chain, starting at the MOEs and ending at Commander's Intent.

In order to score the value for each measure, all of its child measures are retrieved and a weighted average total is calculated. This is done by adding each of the child measures value multiplied by its weight to yield a base score for the measure. The next step multiples the measure's base score by any influences whose conditions were met. This will adjust the score up or down depending on the value of the influence's multiplier. The final step in the scoring of the measure is to limit the score to between 0 and 1. Any score outside of this range is normalized to the nearest threshold.

This process is then repeated for all the measures at each level before proceeding to the next level of measures to score. The end result is a scored COA with a final value at the Commander's Intent level.

3.3 VALIDATION OF PROTOTYPE

3.3.1 Experiment Plan

In order to evaluate the Year III prototype, a scenario was devised where three different COAs were simulated to achieve the desired results. Each of these COAs were scored against the same template ontology and their respective meta-data values to produce scores. Since the COAs were developed to be substantially different from each other, the expectation was that they would generate significantly different scores from each other. These scores were then compared to the actual achievements and losses of each simulation run to see if the generated scores actually matched the human analysis of the COAs respective merits.

3.3.2 Interpreting the Results

After the three COAs were fully defined for the scenarios, the first step was to turn the information for the scenario and COAs into input files for the simulation. After this step was done, the meta-data was defined and added to the FSS input file. At this point, all of the tools and information needed to score the three COAs were present. Figure 10 captures the scores of the COAs.

After the scores were generated, the next step was to analyze the results to ensure that the scores were calculated accurately and that the scoring logic made sense for the scores at each level. One of the surprising things about the score of the COAs was that all of them scored relatively well and that the scores generated were fairly close to each other.

		Score Results	
		Score (%)	Cost of Repair and Replacement
COAs	Overwhelm	89.83%	$10,500,000.00
	Pinion	89.11%	$0.00
	Restrain	96.66%	$2,880,000,000.00

Figure 8. COAs Scores

One of the first things discovered through the results analysis was that each of the COAs was indeed successful in their main objective of preventing WMD from being employed and of disabling WMD production. In the results of each COA from the simulation, the WMD employment sites are disabled by successful missions.

As the analysis of the COAs continued, it came to light that even though the COAs had different means of disabling WMD production and employment, many of the missions that they performed were similar or the same between COAs. For example, all of the COAs had missions to monitor the enemy WMD sites and all had missions to destroy the WMD employment sites. Given that we held the red battle orders constant between the COAs and the same types of assets carried out these missions in each COA, the results were correspondingly similar for all of the COAs. Further analysis showed that the Overwhelm and Pinion COAs were very similar in their mission makeup, the big difference between the two being ops tempo. Where Overwhelm attacks with everything early and fast, Pinion adopts a more metered approach. The Restrain COA was significantly different from the other two than they were from each other and as such yields a unique score.

The next area that was analyzed was the cost of repair and replacement. This was of special interest given the wide variance of costs between the COAs. The Restrain COA incurred such a high cost of repair and replacement as several B52 assets were shot down. The Overwhelm COA utilized several Tomahawk missiles in order to achieve the aggressive ops tempo needed by the plan, resulting in increased cost. The Pinion COA lost no aircraft and utilized no calculated expendables in order to achieve its results, and as such resulted in no cost of repair and replacement. Obviously, it is unrealistic for a COA to have zero cost; this is due to current limitations in the detail of data available. In a real COA, there would be costs associated with use of ordinance, fuel and expendables as well as many other costs not calculated in the Year III prototype. This limitation will be covered more in later sections, but is important to keep in mind when looking at the results.

Even with the limitations on score taken into account, the pinion COA would probably incur the least cost of the three possible choices. Considering the fact that all of the COAs achieve roughly the same level of success, cost might become a determining factor in the COA selection between the three choices. Although the Restrain solution is noticeably more successful, it incurs dramatically greater expense than either of the other two. Given that the Pinion and Overwhelm solutions are very close in their score, the reduced cost of Pinion might make this the COA that would be selected even though it produced the lowest score overall.

4 RESULTS AND DISCUSSION

Automated scenario generation has matured greatly in the last decade. If this trend continues, nearly turnkey scenario generation will be realized. The increase in feasibility of automatic scenario generation increases the corresponding need for automatic scenario evaluation. Dramatically increasing the number of COAs presented to the warfighter for selection will only burden them with too much information. In order to take full advantage of the increase in possible COAs, they must be pruned so that only the best are presented for final analysis and selection.

The best COA may not be the one with the highest score. A COA scoring well against many eCOAs (i.e., robustness) rather than just an expected or most dangerous eCOA may be the more prudent choice. This is especially true when an adversary behaves seemingly erratically, or when confidence in the accuracy of predictions for adversarial actions and reactions is low.

The approaches to developing ontologies supporting robust COA comparison are more varied and subtle than expected. Issues of richness (e.g., depth) versus manageability (e.g., abstraction) are numerous. Additionally, many stated goals are not the actual goals but merely proxies for them, e.g., emphasizing air superiority when the actual goal may be safe supply lines. Careful consideration must be made when constructing an ontology to model score metrics in order to ensure that the proper metrics are emphasized.

Flat data representations such as spreadsheets are very difficult to use and mostly inflexible. This makes such representations resistant to efforts at automation and generally unacceptable for practical application. The user is confronted with too much data, making it hard to focus on the relevant pieces of information and to navigate between related metrics.

The addition of the concept of *influences* to the scoring equation enhanced the ability to model EBO events. These constructs allow specific events to serve as triggers for score modification. By accounting for specific events, the user is able to specify conditions that have a positive or negative impact on the score, regardless of the means by which the events occurred. This mechanism also allows the user to specify sanity checks at appropriate levels in the score, preventing the success or failure of low-level missions from being the sole score determinant if their actions did not produce the anticipated results.

The implementation of the prototype provided many learning points by exposing unconscious assumptions and details not fully developed in previous scoring. This occurred by forcing all of the details and assumptions to be reexamined as part of the automation done by creation of the prototype.

One of the most valuable lessons learned during the development of the prototype was the importance of being able to analyze the COA after it had been scored. After each COA was scored, there was an analysis step done in order to ensure that the score was reasonable given the outcome of the scenario, as well as to refine the COAs and the scoring metrics. While some of this analysis was done for developmental purposes of the toolset, a similar analysis phase would likely be done on a fielded system for both COA refinement and as a score sanity check. This analysis process was found to be labor-intensive and difficult due to the manner in which the Protégé API presented information.

Much of the problem with the information presentation resulted from the fact that the information presented through the Protégé tool is directly coupled to the structure in which the data is stored. For example, instead of being able to show an asset's status as a graph to quickly ascertain what happened to it over the course of the COA, the display was limited to a series of numeric events that reduced the asset's status. Likewise, a network of relationships was reduced to being represented as a series of windows describing the relationships through various pieces of data rather than as text. Ideally, this would be represented through a graphic for easier understanding.

While the Protégé tool provided a very good basic user interface for the prototype needs, the lack of control and customization of display and navigation greatly reduced the usability of the ontology for analysis purposes. The information presented through Protégé remains vastly superior to a spreadsheet approach but still has many areas that could be greatly improved. In order to fully develop the analysis capability of a scored COA, a custom tool should be developed to fully implement the custom display required.

A second key learning point of the prototyping effort was the tradeoff between adding rules and assumptions into the system versus relying on user input. If all values were specified by the user, then the toolset would result in being little more than a calculator to sum the results already specified. By utilizing template values and assumptions, we are able to dramatically reduce the amount of user input required for a score generation and

correspondingly the time involved. This approach must be managed in order to avoid limiting the usefulness of the toolset while still capitalizing off the stored knowledge and timely score generation. A chief goal in rule development should be to focus the user input on criteria that is unique and critical to the scenario being modeled while relying on templates for approximations of common and unimportant values.

A final major lesson learned in the CASA effort was that the score will only be as good as the scoring criteria put into it. While the use of templates and default values goes a long way to being able to generate a score, there is still a need for user-supplied data. Without scoring criteria specific to the scenario being scored, the metrics will only measure how well individual tasks achieved what was set out to be accomplished. The user is required to specify what the relative importance of those tasks is to achieving the higher-level goals, as well as to account for EBO effects on the score using influences.

This necessitation of the user in the loop makes paramount the need to ensure that the data collected is easily assimilated by a user, as well as to provide a clean and understandable interface in which to enter scoring criteria and understand its impact on score calculation. A well-constructed template of default values can be overridden by a single erroneous user-entered value if the weighting is high enough. This level of control is necessary in order to allow the user to correctly model all possible scenarios, but at the same time it puts the burden on the user to ensure the specified criteria is accurate. Because we cannot create a template for every conceivable scenario, we must ensure that the users are able to understand and specify information in as timely a means as possible.

5 SUMMARY

The analysis of enemy behavior and courses of action (COA) are central research topics for military strategists. COAs designed by experts have a need to be evaluated to satisfy the Commander's Intent. However, because the number and complexity of COAs has increased proportional to the complexity of war, a method to automate scoring and evaluation is required. The Course of Action Simulation Analysis (CASA) project's primary goal was to supply military decision makers with tools to formulate and choose the best COAs available.

Development and research of scoring procedures has led to the belief that a COA with the highest score may not be the best in terms of the stated goals. An example would be that destroying many enemy targets would not matter if they still had weapons of mass destruction (WMD) capabilities. Therefore, the ability to analyze COA data and ensure that the proper metrics are being used is crucial.

To effectively evaluate a COA, it was necessary to identify metrics permitting direct comparison of disparate means of accomplishing goals, such as propaganda campaigns versus major interdictions. Additionally, the decomposition of the Commander's Intents into missions was not detailed enough to produce meaningful scores. Therefore, missions were divided into measures of merit (MOM), measures of effectiveness (MOE), and measures of performance (MOP). These measures let the user inspect missions on the asset level and view individual events if needed.

In the final year of the CASA effort, a prototype was implemented and the concept of Effects Based influences was developed. The prototype was tested via the results of three unique COA simulations to be evaluated. These results were parsed from files and populated into a Protégé ontology. Scores were calculated for each COA and influences were attached to illustrate their effects. Influences enhanced the ability to model effects-based operations (EBO) events and were used as checks to make sure conditions in the scenario were met. Different methods to store and display COAs were examined and the ontology data model was chosen. The ability to restructure and manage the data elements and their interrelationships were the reasons to use an ontology data model

This paper will identify in more detail the problem statement, results, conclusions, and future research areas of the CASA system.

6 REFERENCES

[1] Surman, Hillman, and Santos, "Adversarial Inferencing for Generating Dynamic Adversary Behavior," Proceedings of SPIE: The International Society for Optical Engineering Enabling Technologies for Simulation Science VII: AeroSense 2003, Orlando, FL, April 2003.

[2] Fayette, "Effects Based Operations," AFRL Technology Horizons®, Vol 2, No 2, June 2001.

[3] Gilmour, Hanna, and Blank, “Dynamic Resource Allocation in an HPC Environment,” Proceedings of the DoD HPCMP Users Group Conference, Williamsburg, VA, June 2004.

[4] Koziarz, Krause, and Lehman, “Automated Scenario Generation,” Proceedings of SPIE: The International Society for Optical Engineering Enabling Technologies for Simulation Science VII: AeroSense 2003, Orlando, FL, April 2003.

7 ACRONYMS

AFRL Air Force Research Laboratory

API Application programming interface

BN Bayesian network

C2 Command and control

CASA Course of Action Simulation Analysis

COA Course of Action

COG Center of gravity

DCDST Distributed Collaborative Decision Support Technologies

DDD Disrupt, Destroy, Disable

DIME Diplomatic, Information, Military, Economic

DO Delivery order

DoD Department of Defense

EADSIM Extended Air Defense Simulation

EBO Effects-Based Operations

eCOA Enemy course of action

FSS Force structure simulation

GUI Graphical user interface

HPC High-performance computing

IADS Integrated Air Defense Systems

IFS Information Systems Concepts, Applications, and Demonstrations Division

IFSD Collaborative Simulation Technology and Applications Branch

JOPES Joint Operation Planning and Execution System

M&S Modeling and simulation

MOE Measure of Effectiveness

MOM Measure of Merit

MOP Measure of Performance

MRL Multiple rocket launcher

SAIC	Science Applications International Corporation
SGen	Scenario Generator
SGML	Standard generalized markup language
SME	Subject matter experts
SOTA	State of the Art
TOT	Time-over-target
UJTL	Universal Joint Task List
WMD	Weapons of mass destruction
XML	Extensible markup language

ICCRTS 2007

Course of Action Scoring and Analysis

21 June 2007

Chris Egan, SAIC

SAIC From Science to Solutions®

CASA Overview

- CASA is a task for the AFRL to explore ways to score divergent courses of action (COA) while accounting for effects based operations (EBO).
  - Implement a prototype solution for use in further research.
  - COA: taking a series of actions to secure a desired set of outcomes.

EBO

- Effects-based operations (EBO) are actions that achieve the desired results through often indirect and non-obvious means.
  - May often allow for achievement of desired results with less cost or negative results.

CASA goals

- The goal of the prototype was to create a system that could process data from a simulation run and generate a score based on how well that simulation achieved the desired outcome.
- This involved not only evaluating how well individual actions were achieved but also how the actions affected the overall goals.

Scoring Approaches

- Bayesian Networks
- Attrition-based scoring
- Task/effect based scoring

Bayesian Network

- Utilized Netica software from Norsys for the network modeling
- Developed a simple subset of a COA for comparison

Bayesian vs. Ontology

- Pros
  - Easy to see big picture for small data-sets
  - Flexibility to customize scoring using equations
- Cons
  - Even medium sized data-sets become unwieldy
    - No hierarchy to facilitate information partitioning
  - Greater difficulty in analyzing the network to find score drivers
  - Data as compared to information

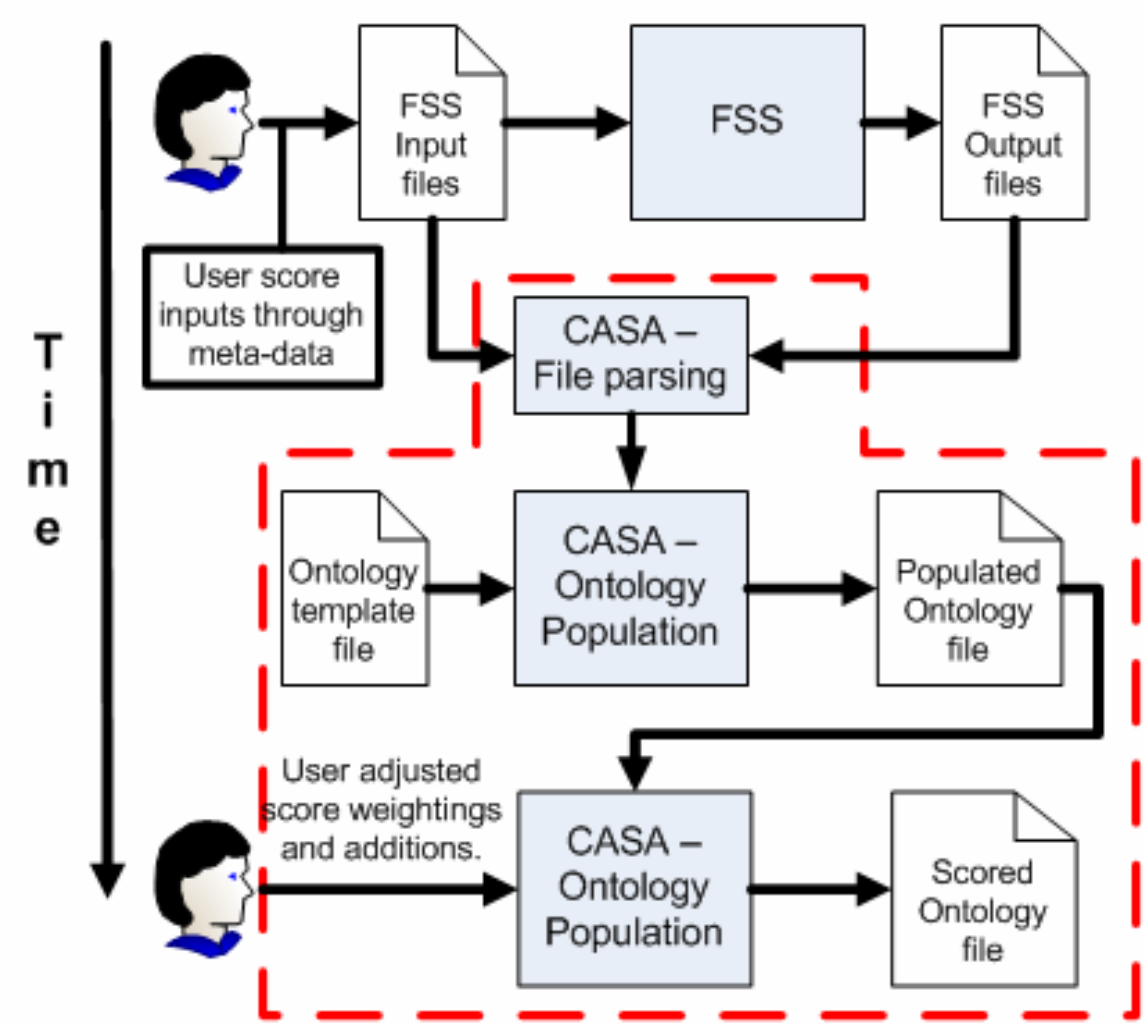
Attrition-based Scoring

- Focuses on the kinetic results of actions, both positive and negative.
- Hard to account for overall desired effects and goals generically.
- There are many exceptions to kinetic force results that are hard to describe generically.
  - Bombing a enemy building is good.
    - Unless that building has civilians in it.
      - Except for when the civilian causalities are merited by the value of the targets disrupted/disabled and destroyed.

Task / Effect Based Scoring

- Adds in the concept of higher level goals and objectives.
- Successful COAs don't just inflict more casualties than sustained but achieve the intended goals.
- Facilitates scoring EBO actions by accounting for overall results.

Prototype Solution



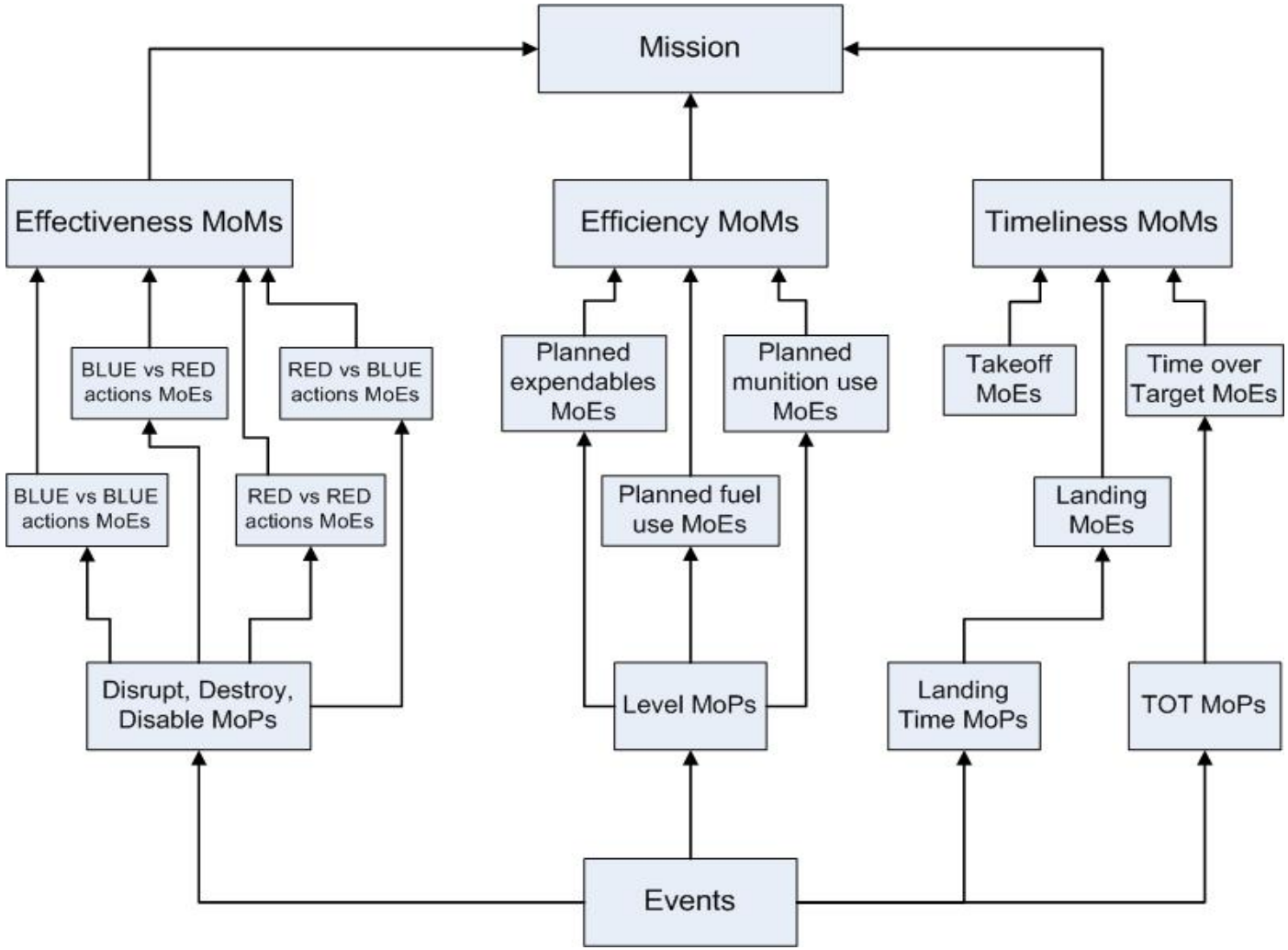
Metrics Breakdown

- In order to facilitate multiple users entering scoring criteria a metrics breakdown was created.
  - Allows levels where a single user can fully understand it's function.
- This also helps facilitate comparison of disparate COAs by breaking down different possibilities into common sets of metrics.

Metrics Breakdown

- Commander's Intent
- Strategic Effect
- Operational Effect
- Operational Task
- Mission
- Measure of Merit: Efficiency; Effectiveness Timeliness
- Measure of Effectiveness: Groups MOPs
- Measure of Performance: Maps to data

Sample breakdown



Scoring MOPs

- After the breakdown has been completed the first step in scoring is to score the MOPs
  - Each MOP will be scored from 0.0 - 1.0
  - A MOP's score is based on how well its actual value achieved the desired value
  - MOPs are populated with data from the sim
  - The desired values come from either meta-data, user-entered values from the GUI or defaults

Score Roll-up

- Once all MOPs are scored, the higher-level scores can be calculated
- Every score-able measure is comprised of a weight and a value
- The score of a higher-level measure is the weighted average total of its constituent measures

Score Roll-up Example

- The mission to bomb the command post has three measures of merit
  - Efficiency MOM: value = .973; weight = .05
  - Effectiveness MOM: value = .793; weight = .7
  - Timeliness MOM: value = .625; weight = .25
  - Value of mission = (.973*.05) + (.793*.7) + (.625*.25) = .76
  - If the weights specified do not sum to 1 they are normalized such that they do sum to 1

Influences

- Mechanism to adjust score outside of the straight roll-up
- Influences provide a means to ensure actions that you wanted to happen did occur and to bias the score accordingly (and vice versa)
- They operate as a multiplier on a specific measure in the score breakdown that is applied contingent upon certain conditions being met

Influences (Cont.)

- If you had an operational task whose goal was to disable a certain center of gravity (communications), you could add an influence to ensure that goal was met
  - Missions could all score well but still leave communications intact
  - Without an influence the operational task would score well even though its true goal was unmet
  - With an influence the conditions would fire and the multiplier could bias the score accordingly

Influences (Cont.)

- Allows users at different levels of the breakdown to identify relationships that they know about without having to fully understand all of the levels in between.
- Influences operate as a multiplier on the score
  - Commutative
    - Ordering not important if multiple influences are applied
    - Useful when multiple users are specifying influences.,

Scoring Assumptions

- One of the driving goals in the prototype solution was to allow for the creation of a reasonable score without specifying every detail in metadata
  - This was achieved through assumptions
    - e.g. destroying civilian non-targetable assets is bad; destroying opposing military targets is good
  - Tried to minimize the number of assumptions while still giving a reasonable score

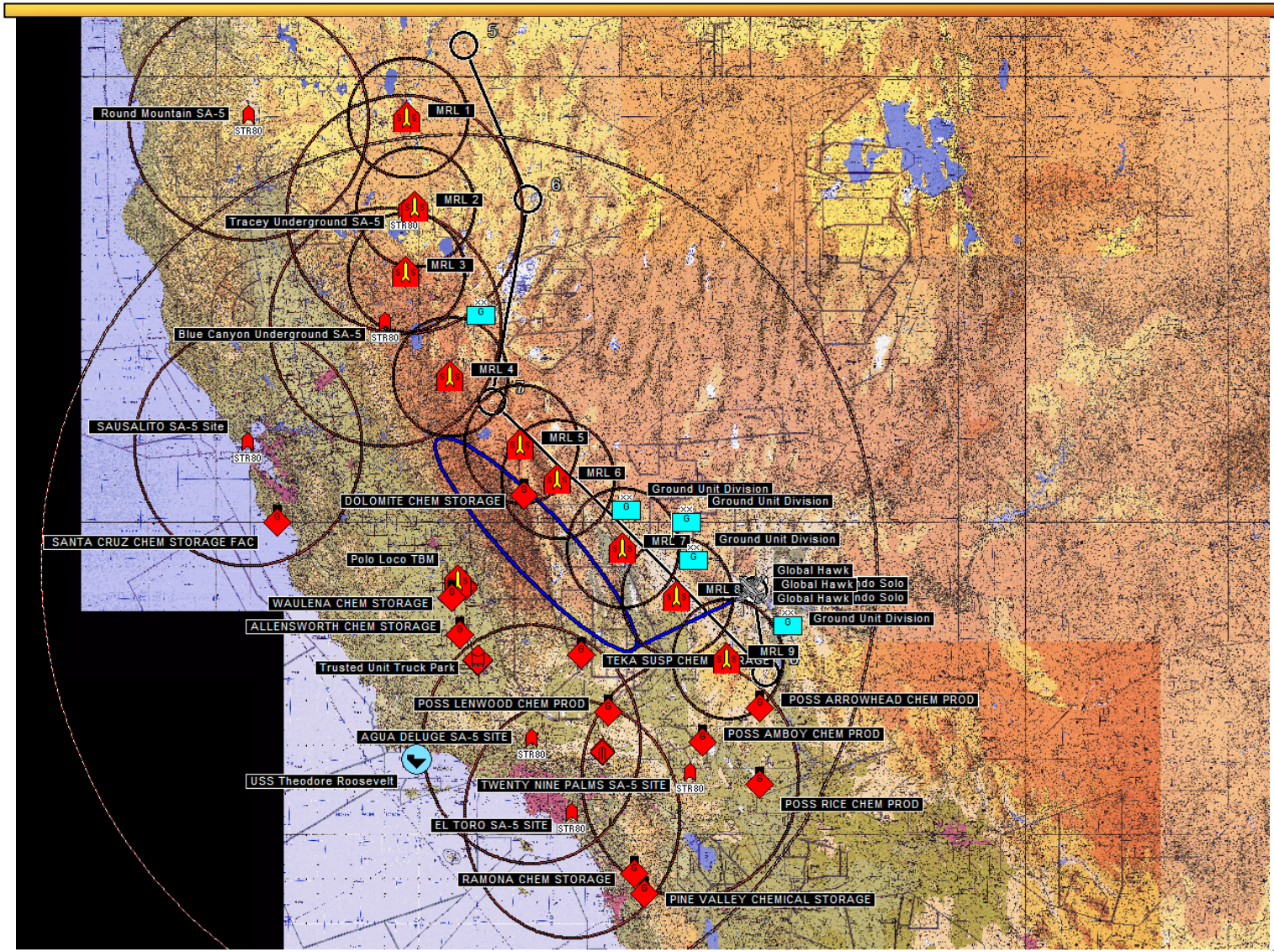
Benefits of using an Ontology

- Captures both the language used in the domain as well as data for specific instances.
  - Extremely useful for sharing information amongst multiple users.
  - Helps avoid confusion from mis-use of terms or differing definitions.
- Allows for inheritance of other Ontologies.
  - Useful for leveraging off of other work and helping achieve data independence.
- Useful for work with living documents
  - Global definition of slots allows for simple propagation of changes throughout the document.

Experiments

- We created a scenario where a red force was mobilizing WMD and conventional forces along it's border in order to negotiate concessions.
- Three COAs were developed to address the threat and eliminate the red WMD threat.
  - Restrain – low ops tempo
  - Pinion – medium ops tempo
  - Overwhelm – high ops temp
- Three commander profiles were created as scoring templates
  - Conservative – high premium on avoidance blue-force and civilian casualties.
  - Aggressive – high tolerance for causalities.
  - Typical – medium tolerance for causalities.

General Force Laydown



Year 3 COA Results

	Conservative	Typical	Aggressive
Overwhelm	0.88735867	0.8983078	0.8766911
Pinion	0.86579984	0.8911031	0.8708272
Restrain	0.78576326	0.9665829	0.8980264

Future Areas of Research

- Score tolerance on simulation results
  - Compile high, low, average as well as standard deviation from multiple seed runs
  - Run the same scenario through multiple simulations to identify and account for simulation limitations and assumptions.
- Score tolerance on User Inputs
  - Run the same scenario with multiple teams to identify the difference between the scores.
  - Repeat the experiment at intervals to determine if the teams vary amongst themselves over time.

Conclusion

- The prototype solution developed provides a basic capability of accounting for EBO effects in COA scoring but much more work remains.
  - Must develop experiments using the tool to identify what works and what requires additional research.

Acknowledgement

The research and prototyping accomplishments of the Course of Action Simulation and Analysis (CASA) task directly result from the vision, leadership, and participation of scientists at the Air Force Research Laboratory Information Directorate (AFRL/IF) in Rome, NY. The authors' efforts were in direct support of ongoing research at AFRL/IF into technology applications aiding military operators in accomplishing assigned warfighting efforts.