Tools For Human Performance Process Modeling and Simulation in Support of Command and Control Analysis

Prepared for the

5th International Command and Control Research and Technology Symposium

Martin J. Brown Jr
BBNT Solutions LLC
9655 Granite Ridge Drive, Suite 245
San Diego, California 92123
(858) 495-2650 FAX (858) 495-7313
mjbrown@bbn.com

Abstract

Command and Control has been, and, for the foreseeable future, will remain, essentially a human centered decision process. Complexity is added because humans, while occupying the central decision making position, must interact with communications networks, computer systems, and organizational hierarchies developed prior to the advent of modern technology.

Most current simulation methods attempt to isolate the human component of the command and control system. Human interactions with the environment are most frequently represented as delays in processing time or as a scripted response following a very simple decision tree. The focus of most simulation environments has been on the technology, communication, organizational, and process issues in isolation. This approach, while facilitating a detailed analytical study of specific aspects of command and control issues misses the forest for the trees. This fragmented approach, separating the organizational, technology and human components of the command and control environment results in a situation where the improvements to a single aspect of the environment are implemented without identifying or incorporating changes in the other facets to optimize the positive impact of these improvements.

Clearly a new approach for analyzing the dynamic interaction among the human, organizational, technological, and communications elements of command and control is needed.

1. Introduction

The Defense Advanced Research Project Agency's (DARPA) initiated a research agenda on the application for the science and mathematics of control theory to the agile control of military operations under the auspices of the Joint Force Air Component Commander (JFACC) Program.

The author wishes to acknowledge the contributions to this research provided by the Defense Advanced Research Projects Agency (DARPA), the Space and Naval Warfare (SPAWAR) San Diego Systems Center, and the Air Force Research Lab in Rome, NY.

The investigation of concepts and technologies for the agile control of military operations called for two complimentary research efforts. The first was the identification and development of a set of Theoretical Techniques and Tools (T3) for the control of military operations. The second effort, and the focus of this paper called for the development of and Air Operations Enterprise Model (AOEM) representing the command and control organizations and processes involved in the planning and execution of air operations. This paper focuses on the processes, technology, and lessons learned by the BBN Technology Solutions team during the development of the Air Operations Enterprise Model. The team consisted of: BBN Technologies, the Artificial Intelligence Applications Institute of the University of Edinburgh, Aptima Corporation, Dynamics Research Corporation, the Laboratory for Intelligent Processes and Systems of the University of Texas, and Zell Technologies Corporation.

The goal of the AOEM is to provide an experimental test bed to support the assessment of various control theories and technologies. This focus on experimental support dictated a component-based enterprise model that could support a "plug and play" strategy for control experiments. This "Plug and Play" requirement was a guiding principle in the design and development of the AOEM.

DARPA's original requirement for the AOEM called for a model of the air operations command and control environment ranging from the JFACC level down to and including the "on board" mission commander. Implied in this statement of requirements was a level of modeling of the opposing forces and of an operational environment to serve as the backdrop for the exploration of command and control dynamics.

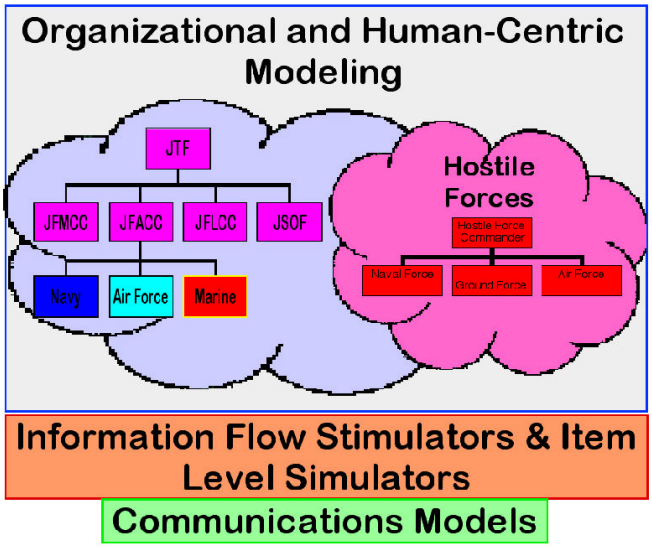


Figure 1: Focus of the AOEM

Three distinct modeling layers were identified as being crucial for AOEM development. The first and most important layer was the Organizational and Human-Centric Modeling. This is the layer that simulated the dynamic interactions among individuals and teams involved in the command and control process. Our representation of the command and control participants (called agents) models planned or pro-active actions designed to support the accomplishment of individually defined goals as well as reactive activities resulting from interruptions in the process. The identification and representation of these reactive triggers proved to be one of the most significant challenges facing the modeling team.

Below this layer the AOEM required a set of integrated Information Flow Stimulators and Item Level Simulators. The primary purpose of this layer was to allow the insertion of external events

to cause interactions at the top layer of the model. These events can be scripted or be dynamically created by internal model agents in response to simulation events. In addition, this layer was used to represent the physical entities in the air operations environment. The representation of physical entities and the separation of ground truth from perceptions provided a second challenge from both a technical and functional perspective.

The third layer of the model represented the characteristics of the communications network supporting military command and control. Defining the supporting communications infrastructure as a separate model component allows us to isolate this important element and then to explore the impact of changes in the communications infrastructure on the command and control processes.

2. **Air Operations Enterprise Model Development Challenges**

The requirements for the Air Operations Enterprise Model posed several significant technical and functional challenges.

Technically, the most important challenge was to identify or create a modeling and simulation environment that supported DARPA's goals for the project. A second challenge facing the BBN AOEM team was to design a model that allowed almost unlimited variations in command and control organizational structures to support the needs of the JFACC System Architect. The third technical challenge was to design the model in a manner that facilitated a number of control theory experiments by allowing individual processes to be removed and replaced by external controllers. The final technical challenge involved developing an interface between the AOEM and the various control technologies being developed in support of the project.

Functionally, our challenge was to accurately describe the dynamics of the command and control environment then to translate the description into a format that could be implemented in the simulation environment. An additional challenge was to organize the components of the model into a structure that captured the true dynamics of the command and control environment while allowing the exploration of alternative organizational and system architectures.

2.1 ***Technical Challenges***

Our selection of a simulation environment was based on our understanding of the requirements for the Air Operations Enterprise Model. We identified three specific types of Modeling and Simulation experiments that the AOEM had to support. The first type of M&S experiment is deterministic: specific inputs produce well-defined outputs. Ideally this should be done much faster than real-time in order to provide efficient services to the technologists. The deterministic model is used to evaluate the impact of experiments and their effect on system stability. The next type of M&S experiment is stochastic: batch evaluation of experiments with Monte Carlo versions of the Information Flow and Item Level simulations. This allows an assessment of controllability of the system. The establishment of appropriate metrics for comparison is critical to analysis.

The last type of M&S experiment is “human-in-the-loop” (HIL). It is often easy to prove by either deterministic or stochastic M&S studies that a new technology will provide major

improvements in a complex process. However, when that technology is implemented and humans have to interact with it, the “dramatic improvements” often disappear and sometimes actually degrade the capability of the humans to make timely and accurate decisions. It is important that the AOEM provide the capability to model HIL. Will the human decision-maker, when presented with a new type of information, better-filtered information, or exotic displays, make better decisions? Only HIL testing can help us answer that question. Stochastic and HIL modeling will be more applicable during final exam experiments.

Our AOEM architecture (Figure 2) fully supported the project requirements. The AOEM architecture follows the High Level Architecture layered approach. At the Platform and Network Layer the AOEM is designed to run on a Windows NT workstation. Our primary interface mechanism employs HLA although a CORBA interface capability is maintained. The heart of the simulation environment is the Distributed Operator Model Architecture (D-OMAR) developed by BBN Technologies. At the Model and Scenario layer we have a set of databases to provide reference and scenario specific data for the model. The Scenario Server, accessed through the user interface modules provides the ability to specify a scenario to run, modify and save scenario variations, and establish exit conditions for the model. The Requirements Integration and Verification Tool (RIVT) from The University of Texas provides configuration management and facilitates per run analysis of scenario and model component configurations. An event and report library provides persistent storage of model results and the Team Integrated Design Environment (TIDE) tool developed by Aptima provides the ability to assess the organizational impact of evaluated technologies. At the top level we provide a set of "Off the Shelf" and custom GUIs for simulation setup, viewing and post run analysis.

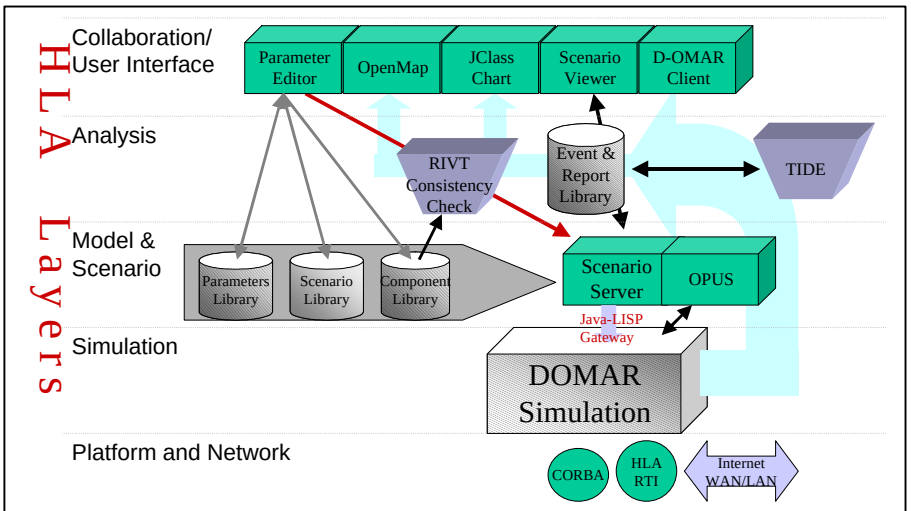


Figure 2. The AOEM Architecture

Several alternative modeling and simulation environments were evaluated before we selected the D-OMAR simulation environment. Figure 3 identifies the criteria we used and summarizes Team BBN’s evaluation of the ability of various types of modeling environments to meet AOEM requirements. For each of the criteria we characterized the performance of each alternative technology. A ‘Full Capability’ rating signified that the specific product/technology fully supported the requirements identified for the AOEM effort. A ‘May Support’ rating signified that the product/technology provided some level of support, or that the assessment was not able to determine supportability. A ‘Not Supported’ rating was assigned only when we were able to

determine that the product/technology could not support the requirements for the AOEM specified by DARPA.

✓ Full Capability ?? May Support Not Supported	Information Flows	Enterprise Modeling	Human-Centered Decision process	Deterministic	Stochastic	Human-in-the-Loop	Distributed	Programming Flexibility
D-OMAR	✓	✓	✓	✓	✓	✓	✓	✓
Gensym G2	✓	✓		✓			✓	✓
Simple++.	✓	✓		✓				
AlphaSim	✓	✓		✓				
Petri-Nets	✓	✓		✓				
Engagement Models	✓	??		✓	??		??	
Wargames	✓	??		✓	??	✓	??	

Figure 3 Alternative Modeling and Simulation Environments

2.1.2. *The Distributed Operator Modeling Architecture (D-OMAR)*

D-OMAR is a discrete-event simulation environment ideally suited to meet the demands of modeling the command and control environment. D-OMAR provides specific capabilities to support the design, execution, and analysis of experiments. The actions and events that drive a scenario are constructed using the same tool set used to build the objects and behaviors of the experiment subjects. A standard set of data collection tools record the behaviors of experiment subjects and timeline displays provide insight into the details of their behaviors. The data collection capability can be extended to meet the requirements of a particular experiment. Links between stimulus and response events can be tracked and missed responses readily identified. Experiment data can readily be made available for analysis using standard COTS analysis tools.

D-OMAR has been used extensively to develop models of human performance and to look at the interaction of human players and software-based technologies. These have often involved models of human players working at a computer-based console. To better understand the processes involved, the interfaces have been developed so that either a human performance model or a human subject can operate them. Data collected on human trials has provided important insights that have contributed to human performance model building.

Core-OMAR forms the basic simulation engine in this framework. The interface to the simulator: the code editors and browsers, the simulation control panel, and the experiment analysis tools, are Java windows that may be called up anywhere in the network relative to multiple instances of Core-OMAR. The simulation control panel provides for scenario selection and the interactive control of the scenario run-time. A trace panel provides scenario-specific information on model execution. A publish-subscribe protocol is the central aspect of the architecture that underlies many of the D-OMAR features discussed here. It operates at the level of a single entity model

serving to coordinate the several goals and procedures that an entity may have active. It plays the same role in the coordination of the activities of several entity models.

At the system level, the publish-subscribe protocol plays in and supports several D-OMAR functions that are important for the AOEM. At the network level, it facilitates the coordination of the operation of multiple copies of Core-OMAR. More importantly, the protocol makes possible the "openness" of the architecture. Using the protocol, D-OMAR readily accommodates the interaction of software components in Lisp, Java, and C++. Working with additional languages is simply a matter of dealing with serialization for the selected language. Hence, the D-OMAR publish-subscribe protocol provides the means to readily integrate products of the technology initiatives.

Another important "openness" issue is the middleware layer used to support distributed operation. D-OMAR has been designed to be largely middleware indifferent and has made use of a variety of middleware layers. In its most straightforward implementations it uses Java RMI, the Common Object Request Broker (CORBA) and the High Level Architecture (HLA).

2.1.2 *The D-OMAR Simulation Construction Environment*

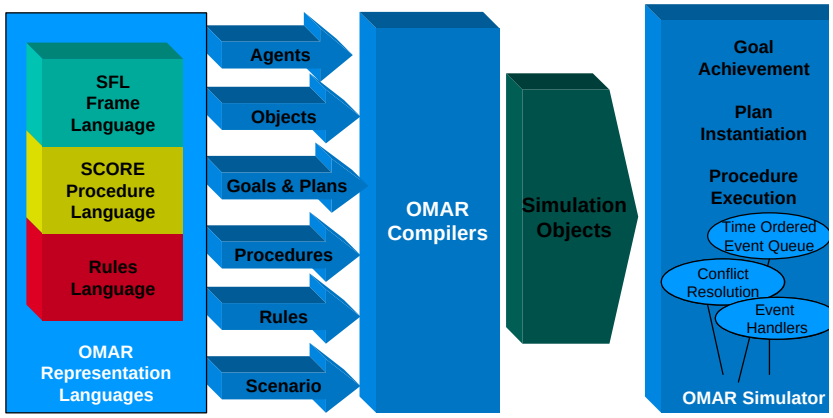


Figure 4. D-OMAR Simulation Construction Environment

The D-OMAR simulation construction environment is shown in figure 4. Construction begins with a set of representation languages for defining rules, procedures and concepts within the model. Using these language representations we construct Agents, Objects Goals and Plans, Procedures, Rules, and Scenarios. Objects represent inanimate objects in the simulation for the most part. Aircraft, weapons, ground units (as a collection of troops and vehicles) and facilities are objects. Agents are a special type of simulation object used to represent decision-making elements within the model. These are referred to as AOEM C2Nodes, and are the model components that interact with the simulation environment. Agent behavior is governed by a set of specific goals, which are achieved through plans. Plans are composed of procedures and bounded by rules. The scenario defines the number and types of agents, objects and the external stimuli that will cause agent proactive and reactive behavior. During execution of the simulation, D-OMAR supports two types of parallel operations. First individual agents can be executing multiple plans simultaneously. The priority for each plan and for each procedure is set but can be modified by factors such as executing procedures have priority over newly initiated procedures by virtue of an inertia function. Contention resolution follows specified rules. The second type of parallel operation supported is the operation of multiple agents each pursuing individual goals. Simulation dynamics occur through the interaction of these agents and through the injection of external events. D-OMAR allows us to characterize and track both proactive (purposeful procedures leading to goal accomplishment) and reactive

(reacting to external agent requests) behaviors in the model. We can also trace a series of events back to the original source event thus establishing a cause and effect relationship.

2.1.3. ***Model Architecture***

Both the second and third technical challenges revolved around the issue of composability. To provide an integrated solution we had to consider interface, functional representations and model granularity issues. Our solution revolved around a C2 Nodes and Process Concept. The C2 Nodes and Process effort began with our functional identification of tasks and activities. We looked for opportunities to consolidate the lower level tasks into product oriented processes. Each process represents a distinct component of the Air Operations Enterprise Model. The products of each process are reflected in the HLA Federation Object Model that provides the basis for our interface with the various Theoretical Technique and Tools (T3) developers working on the JFACC project. We model individual agents to perform the underlying tasks composing the products and then assign these agents to a command and control organization. This organizational affiliation identifies whether internal or external communications models are used to transfer information. The communications model for each organization in the model is explicitly defined. The agents performing a C2 Process inherit the communications model from their organizational affiliation.

Each T3 developed controllers with the intent to “plug” them into the AOEM. We employed the representations of our Federation Object Model for both internal and external interfaces in order to support this "Plug and Play" requirement. A mediator agent dynamically tracks the status of individual model components and determines if the objects being published should be sent over internal or external interface channels. An additional benefit of the C2 Nodes and Process concept from a technical modeling perspective was in controlling complexity. We took a "Black Box" approach to each of the C2 Processes. The complexity of implementation was contained inside the process with only the interface objects visible to the rest of the model.

The fourth technical challenge was to provide an interface that would support a diverse group of technology development teams. Several interface alternatives were evaluated before we settled on the Defense Modeling and Simulation Office's (DMSO) High Level Architecture (HLA). Through this approach we gained several significant advantages by leveraging on HLA's internal house keeping functions (i.e. Time Management and Synchronization) but we also encountered several problems. Management of independent even loops for the simulation, the T3 provided controller and the federation was difficult. Due to the varying degrees of software engineering sophistication among the group supported we provided an Run Time Interface (RTI) helper, a Serializer/Deserializer capability and a Code Generator for creating the federate modules.

The advantages of the HLA interface outweighed these problem areas. A significant advantage in our opinion was the potential opportunities to run the AOEM in a federation with external high fidelity combat simulation models, and logistic models with the AOEM providing a command and control layer and the other models

The final technical challenge was to clearly separate "ground truth" from perception and to provide a user controllable level of uncertainty in the model. Our answer to this challenge was to implement a mediator concept (Figure 5) within the model. Physical entities in the AOEM interact with a unique agent called the "Ether". The ether is responsible for calculating the detection of hostile units and the results of combat engagements. Since our focus was on the command and control decision process we employed a probabilistic model for both detection and combat engagements. For example, an air mission package constantly broadcasts its position to the ether. Surveillance radar and Surface to Air missile sites send out a periodic "Ping" representing the sweep of the radar to the ether. At each "ping" the ether calculates the aircraft in range and then determines if the individual aircraft are detected based on the effectiveness of the radar and the characteristics of the aircraft. The ether sends the result to both the aircraft and the radar/SAM site. The notification will illuminate warning indicators on the aircraft or present a "blip" on the screen. The decision-maker (C2 Node) will become aware of the change in status and initiate appropriate actions based on a modeling of the "cross check" process. Only C2 Node perceptions (information objects) are reflected in the interface for the AOEM.

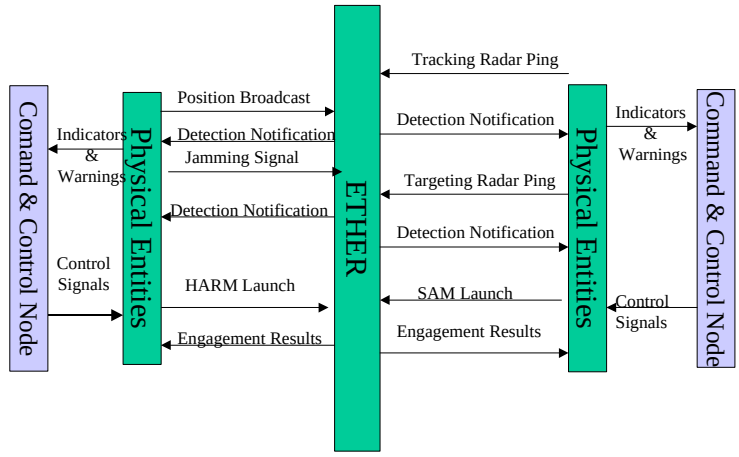


Figure 5. Mediator Concept

2.2 *Functional Challenges*

Our AOEM development process (Figure 6) reflects a multi-level strategy designed to leverage the

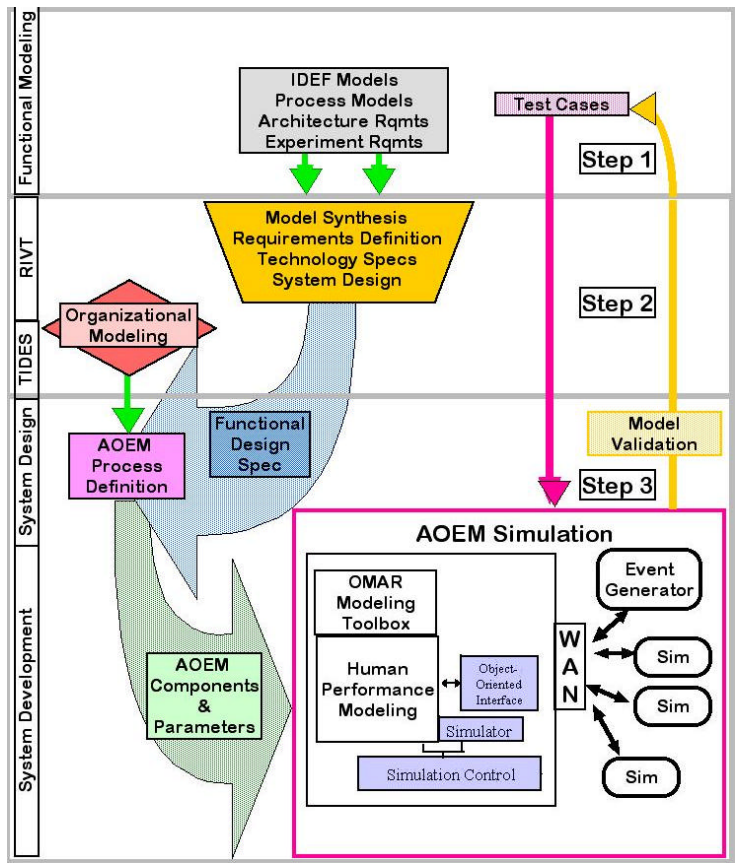


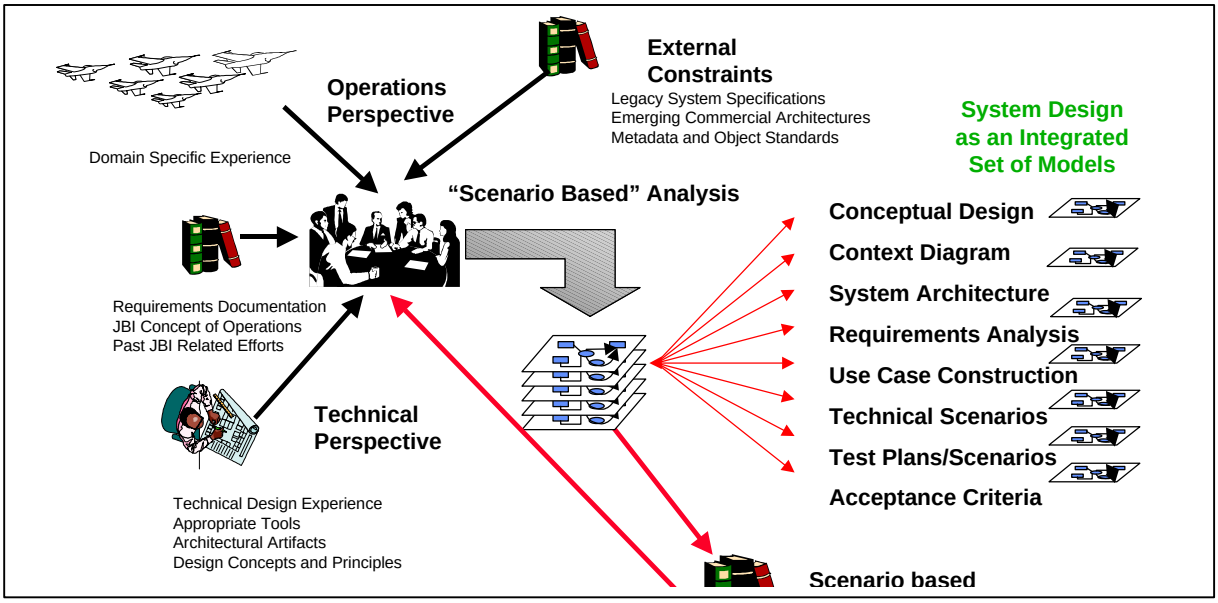
Figure 6: Command and Control Modeling Process

results of previous military modeling efforts, transpose those models into essential elements for design and development of a meaningful command and control simulation, then implement the design in the simulation environment.

We began with a review of existing, related IDEF models, documentation describing the "official" process and command relationship diagrams. We augmented this document description with interviews of command and control domain experts. Our review identified several critical deficiencies in the breadth and depth of the existing models. Currently available IDEF models are focused on the macro-level command organizations. Models exist for the Air Operations Center and the Combat Operations and Combat Plans, but few go into the level of detail necessary to accurately reflect command and control dynamics. Additionally, IDEF models tend to be single process and single organization focused. Links to external processes and organizations are depicted as information flows. We can trace a process through the individual steps within the organization but fail to gain an appreciation for other processes executing simultaneously.

Simultaneous activities are a fact of life with very interesting consequences of interest to human centered decision process modeling. In addition to detailed descriptions of tasks to be modeled we need to address issues such as prioritization, contention, parallel processing, interruptability, and exclusivity. Process aging is also an important issue in an environment where tasks may be suspended in favor of higher priority tasks. The length of time a process can be suspended before it must be re-initialized, rather than resumed, must be determined.

A second deficiency in most current functional modeling methods is their focus on the normal process rather than examining the consequences when things go wrong. In any human centered decision process the more interesting behaviors and stresses on the system are often observed when things begin to go amiss. The problem most often encountered in trying to address decision dynamics under adversity is in identifying the elements that can go wrong. Over the course of interviewing domain experts as part of BBNT's Scenario Template Approach© (Figure 7) the most typical answer when asking what can go wrong is, "It depends." The next question in



these cases is, "Depends on what?" The process is followed until there is a consensus on the three or four most typical factors causing the process into a correction and reaction mode. The result is a set of templates used as the basis for modeling a complex decision process.

Once the elements of the functional model were identified we were faced with the problem of organizing these activities into a functionally coherent, composable structure to support the project goals. The C2 Nodes and Process concept discussed previously fulfilled this requirement and proved to be a bridge between the functional realism and the technical feasibility for the AOEM.

3. **Conclusions**

Development of the Air Operations Enterprise Model is supporting a focused research and development effort into the technologies and procedures necessary for future analysis of the dynamic command and control structure. As we move toward fully integrated environments consisting of human and technology components functioning as a team we can no longer conduct separate analyses of the component of the system. Tighter integration also means that the most significant impact of changes in organizational structure or technology may be felt in areas removed physically but linked through information exchange.

Improvements are needed in the methodology used to conduct functional analysis to capture the types of information needed to accurately reflect system dynamics. Characteristics of procedures such as exclusiveness and contention need to be identified in a systematic manner to support the development of the next generation of command and control simulation models.

On the technology side we need to address the need for federations of models (possibly linked through HLA) to maintain independent as well as shared event loops. Methods to separate ground truth from perceptions and to add controlled uncertainty need to be standardized. Simulation models, constructed as a set of components, are needed to allow analysis of new concepts and technologies. Finally, as these simulation environments increase in complexity, we need the ability to manage the configuration prior to model run time so that we have reasonable assurance that the simulation will proceed without abnormal termination due to missing components or data flows.

5. **References**

[Kott, 1999] Alexander Kott. *Toward a Catalog of Pathological Behaviors in a Complex Enterprise Control System*. Proceedings of the First Aerospace Control Symposium, San Diego, CA, 1999.

[Chen-Burger, 1999] Yun-Heh Chen-Burger, David Robertson. *Knowledge Based Automatic Verification and Validation for Business Models*, Proceedings of the First Aerospace Control Symposium, San Diego, CA, 1999.

[Brown, 1998] Martin J. Brown Jr, *Scenario Template Approach for System Design*, BBN Technologies Internal Process Paper, San Diego, CA 1998

{Deutsch, 1999] Stephen Deutsch, *Distributed OMAR User Programmer Manual, Version3.0*, BBN Technologies, 10 Moulton Street, Cambridge, MA 02138, February 1999.

[Deutsch, 1998] Stephen Deutsch and Nichael Cramer, *OMAR Human Performance Modeling in a Decision Support Environment*, 42nd Annual Meeting of the Human Factors and Ergonomic Society, Chicago, IL, 1998

[Deutsch, 1997] Stephen Deutsch, *Multi-agent Human Performance Modeling in OMAR*, in M. Smith, G. Salvendy & R. Koubek (Eds.) Design of Computing Systems: Social and Ergonomic Considerations (pp. 79-82). Amsterdam: Elsevier, 1997