Models in Distributed C2 Experiment Lifecycles: The Collaborative Experimentation Environment as a Case Study

Anthony J. Bigbee (The MITRE Corporation)

Jonathan A. Curtiss (The MITRE Corporation)

Laurie S. Litwin (The MITRE Corporation)

Michael T. Harkin (The MITRE Corporation)

Abstract

The Collaborative Experimentation Environment (CEE) is a distributed capability and means for designing and conducting joint Net Centric Experiments (NETEXs) where the goal is to explore multi-agency mission effectiveness, whether in national disaster response or responding to in-flight security incidents like terrorist activity. From the inception of the CEE project, the team has used models in deliberate ways across experiment lifecycles, from experiment conception and design, to post-hoc analysis. In this article, our goal is to broadly describe the purpose and evolution of our major model types-- information flow/decision, event response, scenario, domain simulation, data collection and analysis, and architecture. Using this simple taxonomy, we describe each these models and provide examples. We then describe three completed experiments and identify crucial roles and considerations for models within those experiments. We conclude by offering some general lessons learned and identifying future work.

Introduction

This paper uses MITRE's Collaborative Experimentation Environment (CEE) as a case study for the use of models to support C2 experimentation. By models, we mean any artifact—computational or descriptive—shared within the team and with stakeholders that represents a process, concept, or scenario. From the inception of the CEE project, the team has used models in deliberate ways across experiment lifecycles, from experiment conception and design, through experiment execution, to post-hoc analysis. A software engineering analog is model-driven architecture development.

CEE is a distributed capability and means for designing and conducting joint Net Centric Experiments (NETEXs) where the goal is to explore multi-agency mission effectiveness. Each NETEX environment is supposed to reflect real world coordination and collaboration issues where several agencies or organizations (including private and non-governmental) must execute overlapping missions; one example is an in-flight security incident over North American airspace. CEE is intended to support observation, discovery, hypothesis testing, and quantification of effectiveness. Due to the collaborative nature of the project, attainment of shared understanding within the team and between participants about domain knowledge, concepts of operations, and procedures is required.

In this paper, we present a simple taxonomy comprising six major model types.
We then summarize three NETEXs and discuss how models influenced the design, preparation, conduct, and post-hoc analysis phases. We conclude with lessons learned and future directions.

Model Types

The word model has strong semantic connotations for certain communities; we use the term in a broad sense to mean any representation expressed in an artifact intended to be shared or as part of a system used to execute mission tasks in our experiments. Over the course of the CEE NETEXs from 2007 to 2009, we have created six basic types of models: information flow/decision, event response, scenario, domain simulation, data collection and analysis, and architecture. Five of these types are descriptive—and can be construed as conceptual—and the sixth type, domain simulation, can be either predictive, descriptive, or both.

Information Flow Models

Information flow models are meant to depict relationships between potential participants in the domain of the experiment. They are used to explore the domain, scope the experiment, refine hypotheses, and as a reference for other models.

A **Node Information Flow Model** is used to capture a common understanding of the existing real-world relationships between organizations involved in the experiment. This model is created early in the experiment-design process, and is maintained throughout the duration of the experiment.

The model is loosely arranged to show the responsibilities and hierarchical layout of involved organizations. As the NETEXs are collaboration-centric, the model indicates where

collaboration currently exists, as well as the flow of information at a high-level between organizations. For organizations not be present at the start of the vignette time period (e.g., during the initial phases of a crisis), the model indicates when the organization would be stood-up, the mechanics of standing up the organization, and who will have responsibility in the interim. Also indicated, where required, are the locations and availability of resources, where organizations are physically located, if they‘re co-located with other organizations, and other details specific to the experiment.

The model is developed and refined through meetings with internal domain experts and with the organizations expected to take part in the experiment, to ensure that it reflects real-world realities. Often the model is refined in real-time during these meetings.

The example below revolves around the use of Unmanned Airborne Systems (UAS) during crisis response. The node colors correspond to organizations (e.g., FAA, DHS, DoD).

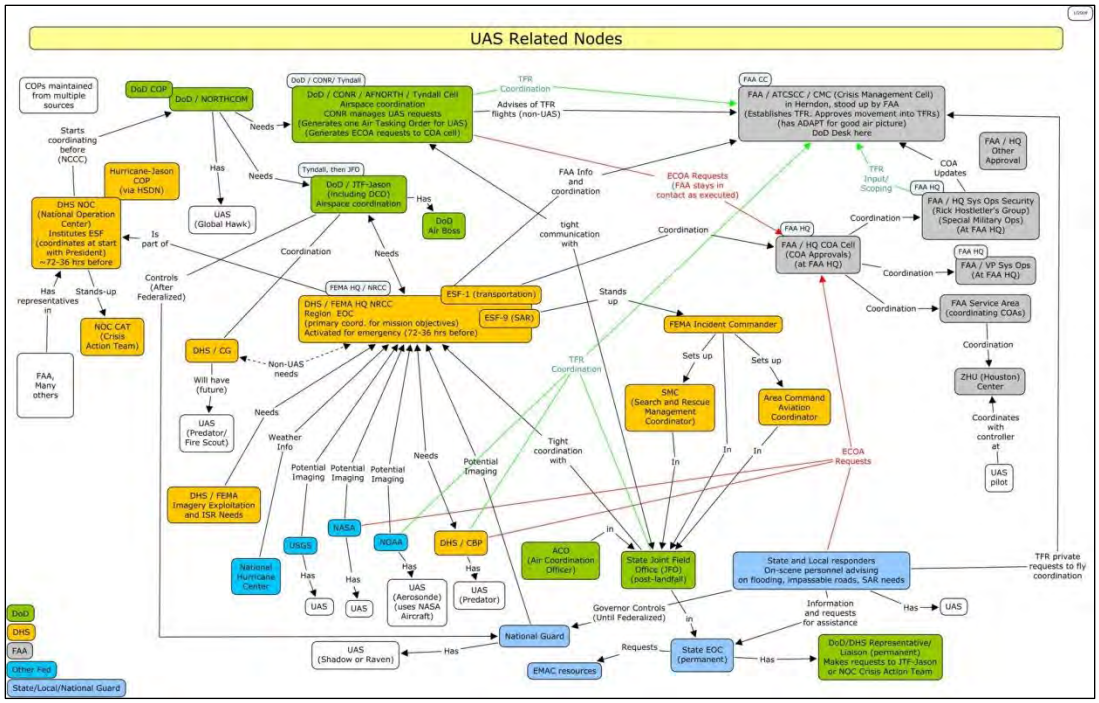

Figure 1. Node Information Flow

The Node Information Flow Model is sometimes rearranged and simplified to illustrate a particular part of the experiment space. Examples include emphasizing the view from a particular organization, examining a particular information flow, or illustrating a timeline for standing-up organizations.

Once familiarity with the domain is achieved, design of the experiment itself begins. A high-level process model is created to depict the overall flow of information, the domain, and decisions related to the experiment. In the case of the Hurricane NETEX (HUREX), we also represented external sources of information from the experiment control cell / the white cell (such as SITREPS). Figure 2 below provide an overview.

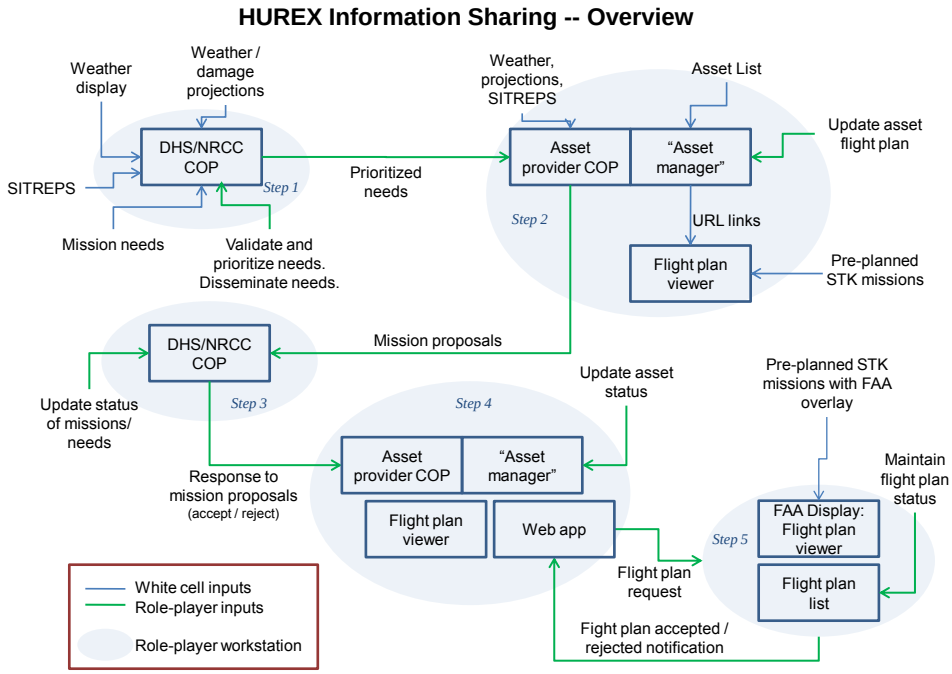

Figure 2. Information Sharing

For each experiment, the team also creates an overall concept of experiment diagram that illustrates the key nodes and functions illustrated by Figure 3 below.

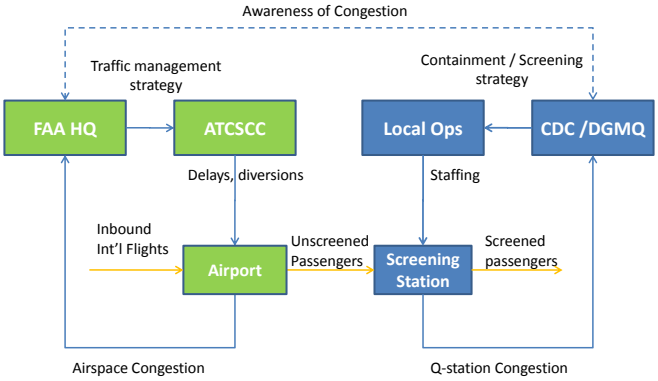

Figure 3. Experiment Concept

Once enough domain knowledge has been acquired and the experiment objectives have been formed, the team may create a participant selection diagram. The purpose of this diagram is to

show key relationships between hypotheses, operational concepts, and participants. It is used to determine which participants from the real world are required and which roles may be played by subject matter experienced MITRE personnel. Figure 4 depicts operational concepts and participants for the in-flight security incident experiment.

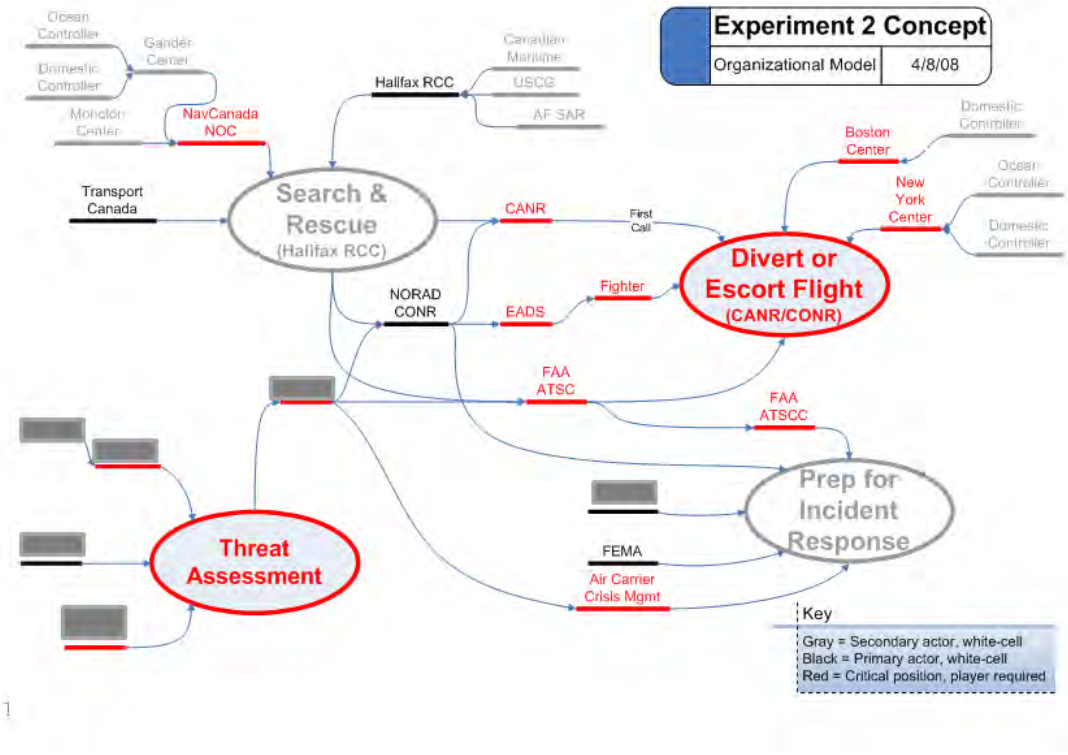

Figure 4. Participant Selection

Event Response Models

With the relationships and communications between the participants well understood, models are built to examine the effects of experimental injections on the participants. **Event Response Models** explore, for each potential inject, the actions the participants are likely to perform. They

indicate, as needed, which roles are being played by the white cell, information necessary to perform the injection, potential actions that may take place because of the inject, and over what medium (e.g. phone, instant messenger, FAX, GRAIL, collaboration tool) the inject and subsequent communications will occur. If necessary, they may indicate where measurements will be taken (to allow for the quantitative analysis) or other parameters important to the experiment. Figure 5 is an example Event Response Model representing interactions that occur for an isolated ill passenger in the PIE experiment.

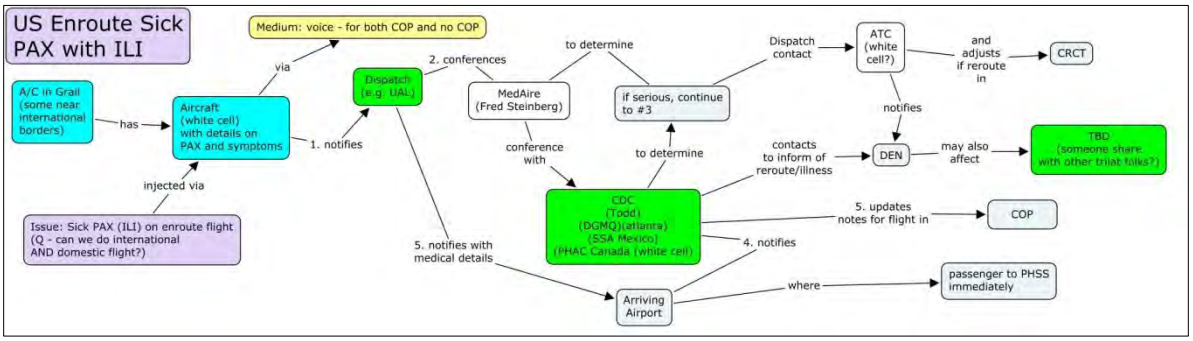

Figure 5. Event Response

Often the experiments require participants to follow a chain of events in order to formulate a proper response. For these, a **Cause and Effect Model** (also known as an Ishikawa or fishbone diagram) is built to track experiment injects and the expected response. An example from 08-02 is shown below in Figure 6.

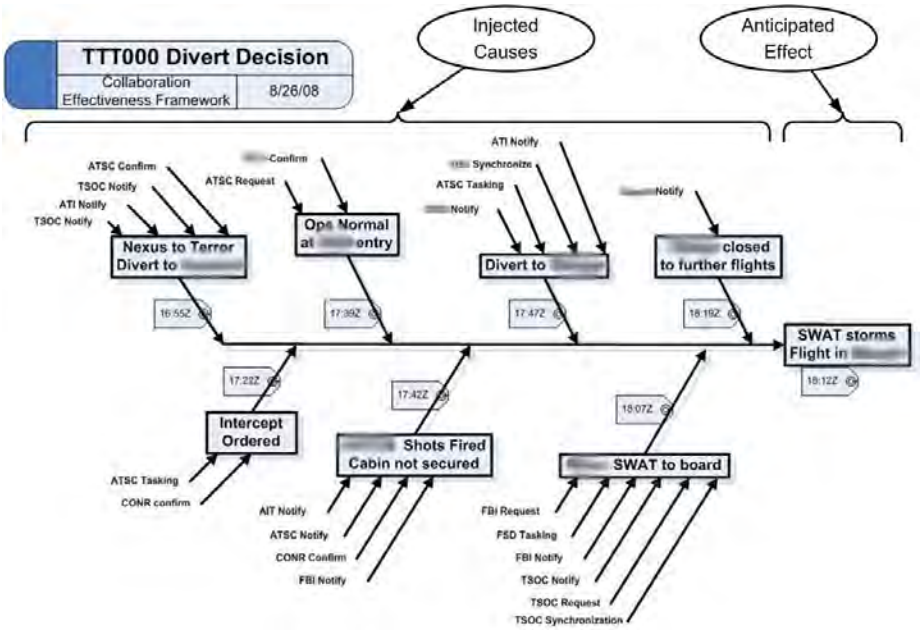

Figure 6. Cause and Effect Model

Scenario Models

Scenario Models form the timeline from which vignette scripts are written. They reflect all injects into the experiment and the expected actions to each inject. For scenarios involving an adversary, scenario models include a backstory, fictitious intelligence (though sometimes derived from the internet descriptions of real-world entities), and other background information deemed necessary.

To build each scenario, the Event Response Models are placed in the order in which they will be executed for each phase of the experiment. The execution of these injects is timed to allow participants the ability to react to the event and deal with it in a manner similar to the real-world-- within the confines of the limited time of the experiment-- and as required to support the testing of the hypothesis.

Figure 7. Scenario Timeline

To assist with envisioning the order of actions and underlying ground-truth in the vignettes, illustrations are often included to help planners and white-cell participants understand what is happening at various points of the timeline. The format of these illustrations is domain-specific. For 08-02, the illustrations were centered on the location of aircraft, and on the complex web of terrorist relationships to each other, to other assets, and available intelligence information. For the HUREX, illustrations showed the hurricane track, mission-needs, missions in progress and available assets at various times during the vignettes. An example of a HUREX illustration is shown below in Figure 8.

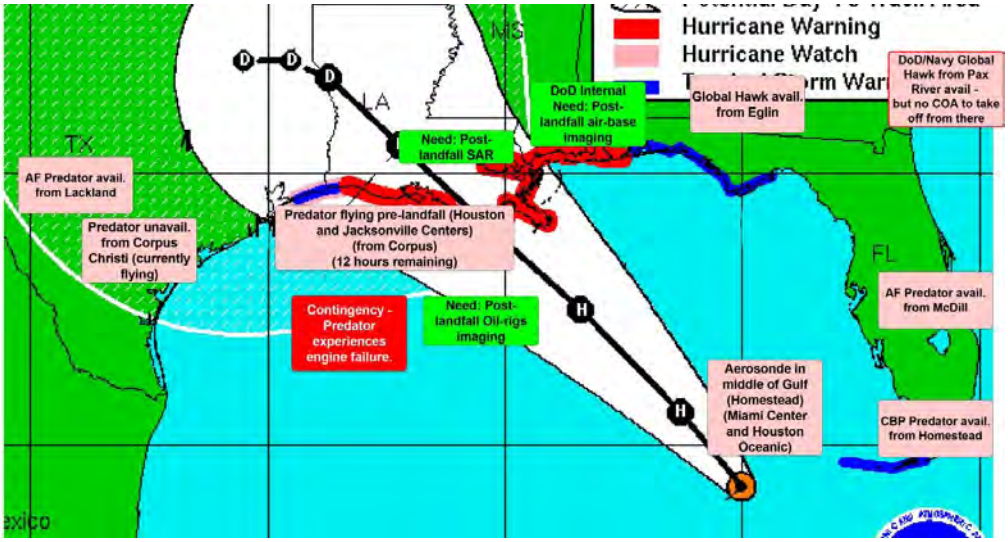

Figure 8. Scenario Illustration

Scenario models (and event response models) also reveal the content and ideal timing of event injects and artifacts needed for the experiment. NETEX injects may be voice (via telephone or teleconference), email, email attachments, text messages, automated systems (e.g. GRAIL), or other novel means (e.g. a collaboration tool). For the HUREX, artifacts included hurricane reports from the National Hurricane Center, weather reports from the National Weather Service, and SITREPS (situation reports) from state and local governments.

Experience has shown that by rehearsing the participant actions via the scenario models with a _what-if‟ attitude increases the likelihood that unanticipated, participant-generated, events will be handled appropriately by the white cell. Additionally, by having the models (and artifacts) on hand, the white-cell is better able to respond to the inevitable unanticipated event, and better respond to participants pulling the information rather than waiting for the information to be pushed to them at the scripted time.

Lastly, having the scenario models available during the experiment can help focus the attention of the white-cell to keep the important aspects of the experiment on-track.

Domain Simulation Models

Epstein (2008) identifies 16 reasons to use domain simulation (i.e. computational) models for purposes other than prediction. The team uses simulation models when domain aspects are:

- Complex -- many elements, non-linear outcomes and effects, many relationships, many different kinds of elements)
- Dynamic – change is frequent and/or difficult to script
- Critical to the goals of the experiment – participants need a realistic enough representation of the domain

We generally ask these questions – can the decision and coordination stimuli be provided by _scripting' kinds of inputs or not? What level of fidelity is required? If the answer to the first question is no, a search for GOTS and COTS follows. If no suitable simulation models are available (cost-effective, adequate data interfaces, and so forth), the team considers developing one from scratch. This generally requires a process that is discrete enough to be modeled and adequate existing domain knowledge.

Further empirical domain knowledge must be acquired to set parameters appropriately. We conduct literature searches and interviews with experts. This may mean identifying abnormal or extreme values that would not normally occur in order to test some aspect of the hypothesis. An initial reaction from stakeholders/SMEs can be ~~"t~~his would never happen."; but, see Kliemt

(1996) for a discussion about the utility of ―unrealistic" models and ―radical" assumptions. Balancing parameter values across experimental conditions is a challenging process. Finally, one of the significant outcomes of our experiments is the identification of gaps in existing knowledge for setting parameters.

Simulation models are integrated into the experiment environment by either wrapping them with basic decision support capabilities, or connecting them with sensor models or C2 tools which expect certain kinds of input. Detailed discussion is beyond the scope of this paper, but integrating any kind of simulation into the experiment environment often requires significant resources. This is often a high-risk area that should be dealt with as early in the planning and integration process as possible and by experienced simulation engineers.

In experiments conducted to date, we have brought to bear several different kinds of simulation models:

- Spreadsheet-based
- Numerical Computing Environments such as MatLab
- Entity and Agent-based
- Combat and Sensor

Across our experiments, we have usually used a combination of entity/agent-based models and combat and sensor models. For example, in NETEX-08-01, we use the Joint Semi-Automated Forces Simulation (JSAF) and Airborne Warfare Simulator (AWSIM) to simulate the behaviors of surveillance radars, and MITRE's GRAIL Real-time ATM Infrastructure Laboratory (GRAIL) simulation to represent airborne aircraft in the National Airspace System (NAS). MITRE

developed custom software to integrate these tools using the Distributed Interactive Simulation protocol. In the Pandemic Flu experiment, we developed an airport screening model from scratch using the NetLogo agent-based modeling environment. The team has experience with other agent-based modeling toolkits such as Repast and MASON, but the airport screening modeling chose NetLogo for its visual integration and other features that support rapid prototyping. One advantage to developing a simulation from scratch is that input to and output from the simulation is completely controllable.

Domain models are also useful during experiment planning and design and post hoc analysis. In our most recent experiment involving airport screening for pandemic flu, the airport screening model developed by the team played a crucial role in all phases of the experiment. The screenshot below in Figure 9 depicts a day‘s worth of screening activity at an airport and graphs portraying delays throughout the process.

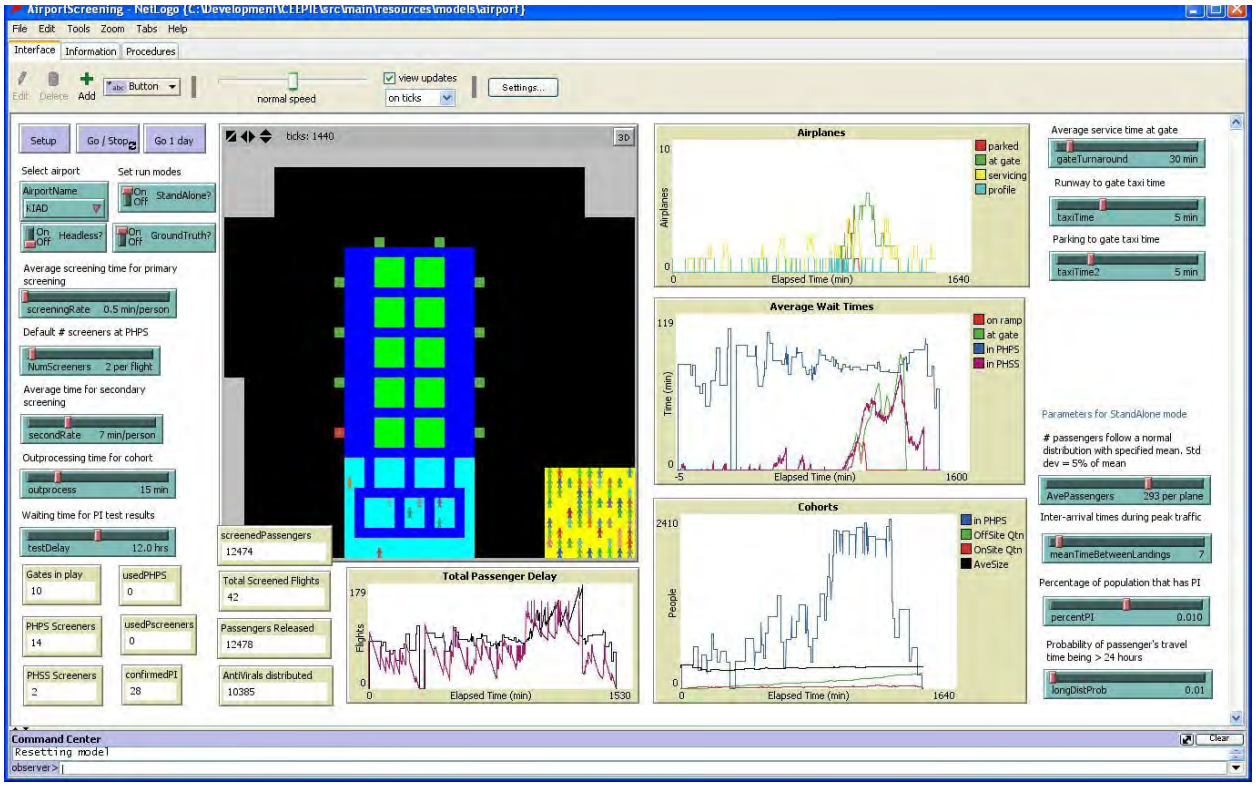

Figure 9. NetLogo Airport Screening Model Screenshot

Data Collection and Analysis Models

As experiment design and preparation unfolds, the data collection and analysis team begins creating the data collection and analysis plan. Typically, that team relies on or builds upon the last experiment's approach. The meta-model below illustrates the key issues in data collection and analysis. In general, a triangulation approach must be used as the experiments feature cognitive behaviors such as decision-making/judgment—no single measurement technique or approach would be sufficient.

Architecture Models

Architecture, or system, models represent the systems and components in the experimentation environment. While there are a variety of system or technical modeling standards, such as

UML, SML, IDEF, and we use simple block diagrams to represent (sub)systems and certain kinds of interactions or data flows. As C2 architecture modeling has received a great deal of attention in the last two decades, particularly in software engineering, we conclude this subsection by providing one of our architecture model in Figure 10 below that characterizes interprocess communication between components.

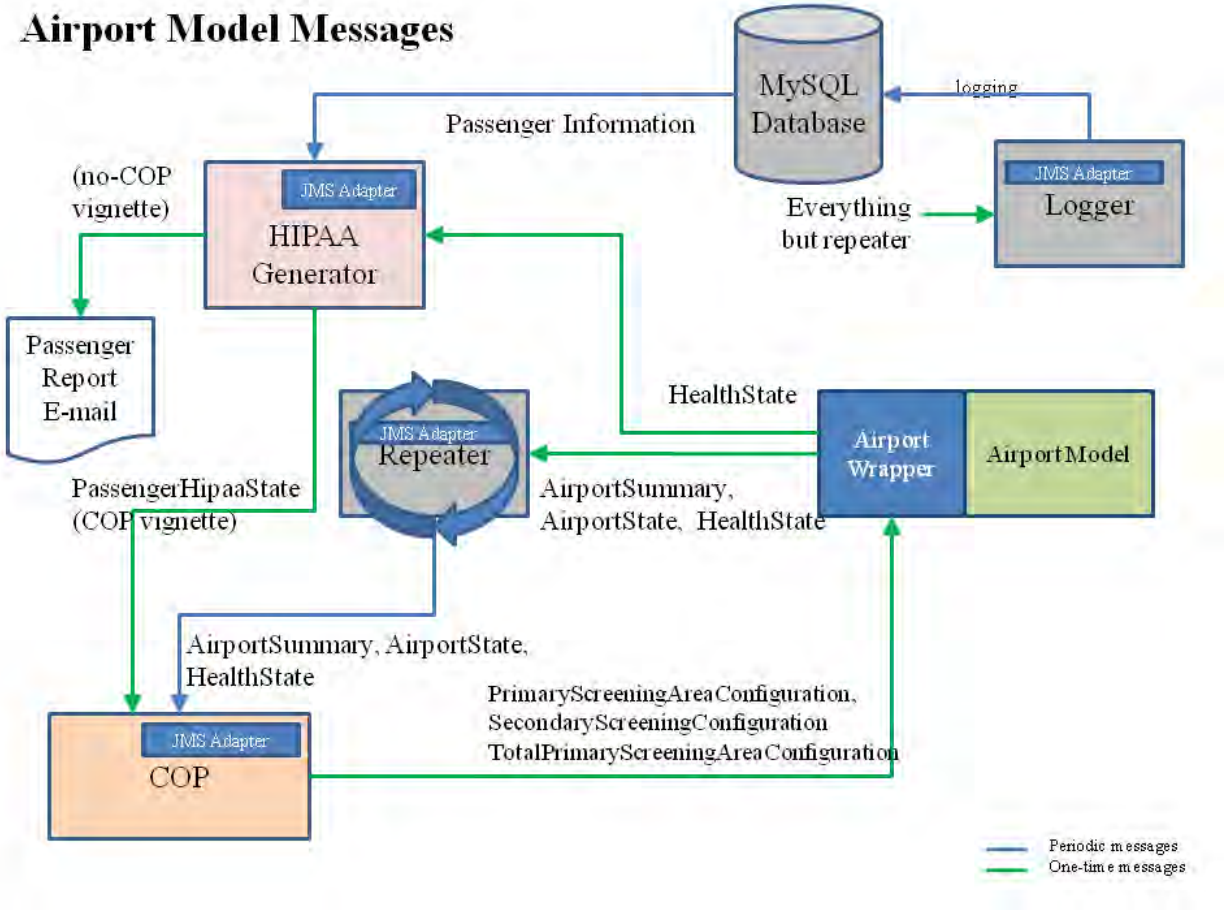

Figure 10. Interprocess Communication

Model Forms and Standards

At this point, it is worth describing the concrete forms used during creation and sharing. We typically share the artifacts as either Adobe Acrobat or Microsoft PowerPoint files. During

construction, we use iGrafx, IHMC CmapTools, Visio, or PowerPoint. The first two tools have the most specific functionality for creating different kinds of diagrammatic (and executable) models, as they impose structure and rules on the layout of objects in the model. They are not widely used, however, which can inhibit collaboration under some circumstances. Visio and PowerPoint are more generic and widespread, so are better for enabling remote collaboration. We typically generate Acrobat or other kinds of common file formats to share within the team or to share with stakeholders. Computational model forms are used in specific ways by computers and we generally retain them in their native form.

Although members of our team are familiar with standards such as IDEF and UML, we generally do not use them because of time constraints, specialized tools required (though free software tools do exist), and those standards focus on software engineering. Some of our models, however, resemble specific types within those standards, such as a UML sequence diagram.

Experiments

In this section we use three experiments as cases for how and why models benefitted the experiment phase. Each experiment represents different domains, although all three feature the air domain and FAA operations.

NETEX 08-02 In-Flight Security Incidents

In-Flight security incidents may result in mission execution requirements for many organizations. In a potential terrorist seizure of an aircraft bound for North America, for example, law enforcement, air traffic control, military air defense, intelligence community,

emergency response, commercial air carriers, state and local government, and others may become involved.

In this experiment, it was critical to represent a significant amount of airborne traffic over the North Atlantic and the U.S. and provide ‗noise' events for the participants. In part, this was due to the terminal response phase of the incident where aircraft were judged hostile and tactical engagement decisions had to be made. Additionally, the joint collaboration space represented by the Domestic Events Network (FAA, 2009) meant that representing the roles and responsibilities of each organization were important to build consensus and solicit participation prior to the experiment.

Since participants are generally allowed to behave creatively during the experiment, one of the challenges the team faced was to identify likely and significant courses of action and decisions participants would take. An outcome of scenario modeling and development is creating events that channel and create situations to meet experiment goals and to provide options if participants deviate from expectations (which usually happens). Participants' creative responses are desirable and sometimes the vignettes do unfold in unexpected ways, but the overall goals of the experiment must be met. When experiments involve a thinking adversary, event response models and scenario model artifacts are important to have in the experiment control/white cell to allow the white cell to deal with unexpected participant decisions.

NETEX 09-01 UAS for National Crisis Response (HUREX)

The HUREX was conducted to evaluate alternative concepts for UAS usage and coordination in a national disaster response. With many possible assets and asset providers, and only one UAS

allowed to fly per FAA facility, the experiment looked at how to coordinate and optimally allocate UAS assets owned by different chains of command (DHS versus DoD) and how to keep the FAA in the loop (Maroney et al, 2009).

In this experiment, the phenomenon driving the situation was relatively slow compared to human decision-making and time available for each experiment vignette. As a result, the team made three major design choices. First, no domain simulation model would be required; the scenario team would be able to create scripted weather and state/local sitreps, sensing collection requirements, and assets available. The weather, flooding, and airborne inventory did not require explicit simulation. Secondly, to avoid the cost of acquiring and integrating flight planning tools, the team scripted flight plans in order to provide asset providers with enough options to meet collection requirements. Finally, to examine behavior during different phases of a weather disaster—pre-landfall planning, landfall situations, and post-landfall response, each vignette featured a time ‗jump' where time was advanced by 12 to 24 hours and they were provided with new situation updates. Since no simulations were used, only the MITRE-developed prototype decision support tool required clock ‗advancement.' Time shifting within a vignette when multiple simulations and applications are part of the experiment environment is usually difficult at best.

NETEX 09-02 Pandemic Flu Experiment (PIE)

This experiment featured a return to using GRAIL to simulate the NAS and also included a novel simulation—the airport screening model. There are many potential aviation impacts to implementing a Risk-Based Border Screening Strategy (Center for Infectious Disease Research

and Policy, 2008) for detecting and mitigating pandemic flu in the U.S., and the process itself has never been implemented. Given both of these factors, MITRE implemented an airport screening simulation model. The behavior of this model and eliciting required parameters elicited significant discussion within the health and aviation community experts that CEE team consulted. It served to clarify requirements and also enabled integration with the ground truth model (when passenger carrying aircraft landed). A joint research project—MITRE and RTI—provided a global disease spread model. This model was used to seed arriving international flights with passengers having the flu, influenza-like illness, and other parameters. This model was combined with GRAIL to create the data necessary for the airport screening model. During the experiment itself, the airport screening model provided the ground truth for where in the screening process any particular international flight arrival at any of the designated U.S. screening airports (aka ‗Q Stations'). Some passengers would be onboard awaiting screening, some would be in primary screening, and sometimes, a small number would be in secondary screening. The output of the model was fed to a prototype decision support tool that enabled participants to assess delays within and across the airports, predict future delays, and take actions, such as assigning more screeners to a flight in screening.

Finally, the airport screening model is being used during post hoc analysis to address:

- Compare participant behavior with other possible courses of action
- Identify impacts to overall screening times and delays given low background influenza-like illness rates and high-influenza
- Identify if reactive behavior is sufficient to prevent delays from becoming worse or if proactive behavior and predictive decision support is necessary

Lessons Learned and Future Work

Lesson Learned

Our experiences across several experiments have enabled us to make the following judgments and lessons learned:

1. Models have value across experiment lifecycles even if they are designed for only one phase. The clearest example is when post-hoc analysis is conducted. The analysis team had all the modeling artifacts created prior to the experiment and could compare data collected with intended events and processes.

2. Unless descriptive models are created in one tool, there is little to no model _interoperability' or way to automatically link data collected with concepts and relationships created in the modeling tools.

For example, there is no way to easily construct a _query' that would trace information created by one role to other roles that should have used that information as defined in a model. For certain kinds of experiments, the ability to quickly construct and execute these queries would make analysis more efficient and might lead to more or refined insights.

3. Models are useful stimulants for eliciting discussion. When a process or concept is concretely described by diagrams, stakeholders and participants become willing to provide clarification, ideas, or pose questions. Enabling interaction between stakeholders is one of the primary objectives of the CEE project.

4. Models do not take the place of active team communication and consensus building. With any written artifact, it is easy for authors to assume that content will be immediately considered and understood. Team members have multiple project responsibilities, are in different locations,

and have different responsibilities. Active promotion of shared understanding via periodic meetings to review content is still required.

5. Choice of modeling tool matters. For descriptive models, it is often the experience and familiarity with a tool that influences a modeler‘s choice. One result of this choice is which kinds of descriptive models can be brought together in that tool. For domain simulation models, the impact is potentially more significant due to integration and interoperability requirements. Typically, this choice should involve the integration and architecture team(s).

Future Work

The team is always looking for ways to rapidly create, modify, and share models. When scope or experiment designs change, the work necessary to refine models is unavoidable. We would like to capitalize on completed modeling during post hoc analysis, however, as discussed in the lessons learned. We believe that semantic technologies are a key part of the solution. A complete solution might include:

- Usage of ontologies by all models-- the use of standard terms of references (concepts) with associated attributes, relationship types (arcs), and in one or more ontologies
- Data collected and tagged using the same ontologies
- Intelligent queries on collected data using a semantically-enabled database. The technical language of the queries would be hidden by user interfaces
- The ability to pose queries and set alerts in near-real time for data collectors to increase their situation awareness and improve their judgments

Acknowledgements

We are grateful for the expertise and time of our government and industry participants. We also thank our CEE teammates for their insights, as well as MITRE experts who lent their advice during planning and performed white cell roles during the experiments. This work was funded by internal MITRE research and development.

References

Epstein, Joshua. 2008. Why Model? Journal of Artificial Societies and Social Simulation vol. 11, no. 4 12. http://jasss.soc.surrey.ac.uk/11/4/12.html.

Federal Aviation Administration. 2009. Domestic Events Network (DEN) and Presidential/United States Secret Service (USSS)- Supported Very Important Person (VIP) Movement. Notice N JO 7210.724. http://www.faa.gov/regulations_policies/orders_notices/index.cfm/go/document.information/documentID/99629.

Kliemt, Hartmut. 1996. Simulation and Rational Practice. In *Modeling and Simulation in the Social Sciences from the Philosophy Point of View*, ed. Rainer Hegselmann and Ulrich Mueller, 13-28. Dordrecht, Kluwer Academic Publishers.

Maroney, David R., et al. 2009. HUREX – Experiment of Use and Prioritization of UAS in Hurricane Disaster Response. Paper presented at the 9th AIAA Aviation Technology, Integration, and Operations Conference (ATIO), Hilton Head, South Carolina, 21-23 September 2009.

Schnirring, Lisa. 2008. Feds Stage Airport Test of Plan to Slow Pandemic. Center for Infectious Disease Research and Policy. Minneapolis. http://www.cidrap.umn.edu/cidrap/content/influenza/panflu/news/nov1208airport.html.