Approved for Public Release; Distribution Unlimited
Case # 07-0455

Hybrid Systems Dynamic, Petri Net, and Agent-Based Modeling of the Air and Space Operations Center

Jennifer Mathieu, John James, Paula Mahoney,
Lindsley Boiney, Richard Hubbard, and Brian White

The MITRE Corporation, 202 Burlington Road, Bedford, MA 01730

©2007 The MITRE Corporation. All rights reserved.

Abstract

In an earlier paper by the authors, an existing Air and Space Operations Center (AOC) process model (i.e., Petri net) and new global and mission models for the environment in which the AOC operates (i.e., System Dynamics) were linked (federated). The focus of this paper is the development of an operator-environment model (i.e., Agent-Based Model). An existing systems framework for attention allocation of operators within the AOC has been implemented in AnyLogic® software that supports multiple modeling paradigms. Subject matter experts have helped define operator interruptions (defined as shifts in activity), involving: 1) the agent's past experience with the source (i.e., person) of the information; 2) the information type (e.g., change priorities or distractions); 3) the communication modality (e.g., Chat, E-mail, telephone, or in-person); and 4) whether attention allocation attractors (e.g., information filters, alerts, or summary displays) are present. The results for linking (federating) the Petri net and System Dynamics models are summarized, and new results for the Agent-Based Model are presented based on a pilot-down scenario. It has been observed that many AOC operators can become distracted by a pilot-down critical event, even if the operator is not able to directly assist in the rescue. Furthermore, this distraction has been hypothesized to have a detrimental effect on the activities the non-involved operators are currently handling. Linking the Agent-Based Model to the Petri net model is our next step towards a multi-scale, hybrid model of the AOC.

Background

A set of inter-related activities (or Regimen) for Complex-System Engineering has been suggested (Kuras and White, 2005; White, 2005; White, 2005a; Kuras and White, 2006), including:

1. Analyze and shape the environment
2. Tailor developmental methods to specific regimes and scales
3. Identify or define targeted outcome spaces
4. Establish rewards (and penalties)
5. Judge actual results and allocate rewards
6. Formulate and apply developmental stimulants
7. Characterize continuously
8. Formulate and enforce fitness regulations (policing).

These activities were hypothesized to focus and accelerate the natural evolution for the potential benefit of the system. The goal of this effort is to develop a multi-scale, hybrid model using real data and subject matter experts (e.g., Figure 1). The main research question is how to link (federate) the models from the various scales together. Once a multi-scale model exists, it will be possible to more fully apply the activities listed above. As part of this study, analyzing and shaping the environment, tailoring development methods, judging results, characterizing continuously, and enforcing fitness regulations were all part of the modeling process. The remaining methods can be applied after the multi-scale model is validated.

In an earlier paper, Mathieu *et al*. (2007), used an existing Air and Space Operations Center (AOC) process model, and developed global and mission models for the environment in which the AOC operates (Figure 1). The models were developed in separate software, MSim and Vensim® respectively, and the main challenge was linking the models. The Petri net, AOC process model used discrete time while the System Dynamics, global and mission models used continuous time. In addition, the Petri net model was validated using data over the course on 1 day (1990 Gulf War), and the System Dynamics model was designed to run over a period of 60 days. Finally, model elements were needed to link the Petri net and System Dynamics models, and these were created using Vensim® because of its rapid model development capabilities. However, this required the re-creation of partial abstractions of the AOC model in Vensim®. The results of this analysis will be presented briefly below.

The operator-environment model is the focus of this paper. AnyLogic® was selected as the software environment in which to build the Agent-Based Model—this environment can support multiple modeling paradigms in the same tool, thus addressing the difficulty in linking the models. AnyLogic® has the following features which facilitates multi-scale, hybrid modeling:

- Supports System Dynamics, Discrete-Event, Agent-Based, and Dynamic Systems modeling while using the same timing engine
    - Maintains discrete or continuous space
    - Easy to adjust the level of abstraction
    - Possible to choose the best approach for the problem
    - Easy to switch from one approach to another
    - Easy to mix approaches to develop better models
- Based on Java, Object-Oriented
    - Viewer/Debugger
    - Standalone Java Application, Java Applet, Models in XML format
    - Java Remote Method Invocation (RMI), High Level Architecture (HLA)

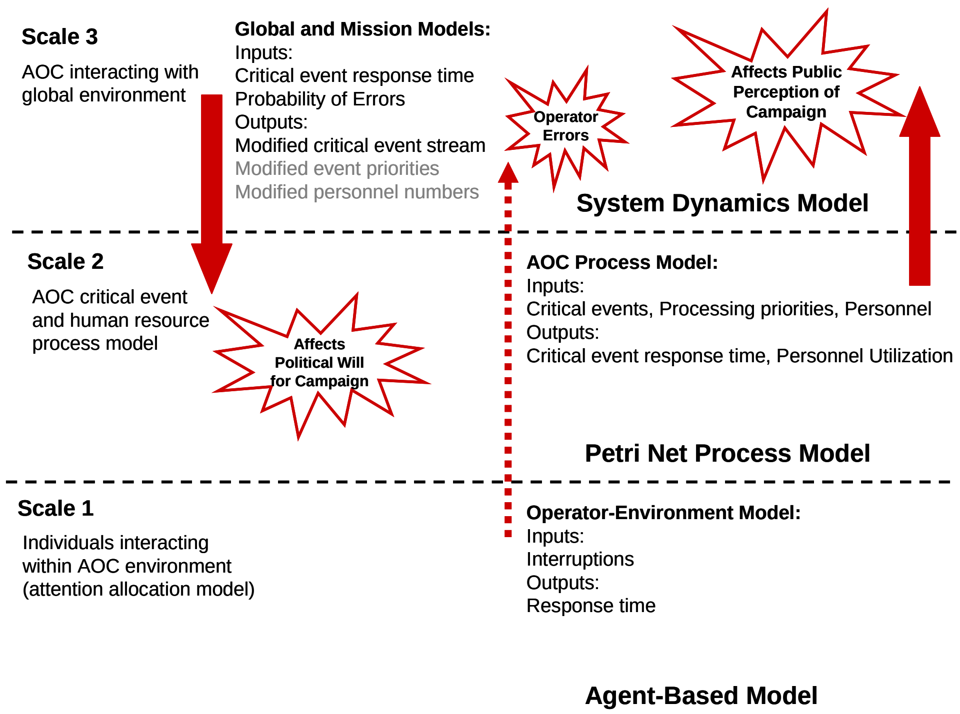

Figure 1. Multi-scale, hybrid model of the AOC with possible linking method.

Operator-Environment Model (Agent-Based)

Boiney (2007) describes a systems framework for attention allocation of operators within the AOC, as shown in Figure 2. This framework was developed using direct field observations of operators in Command and Control environments (e.g., Joint Expeditionary Forces Experiment, JEFX '06). The focus of the Agent-Based Modeling effort is the Attention Allocation System shown in the center of the figure. In this first iteration of the model, agents are assumed to focus on only one activity at a time. Therefore, the Primary Attention and Secondary Attention activity boxes shown in Figure 2 are modeled as a single operator focus. The various activities that the operator must handle have priority and difficulty (e.g., related to stress) levels and can spawn additional self-activities (e.g., related to memory) such as completing a checklist.

This interpretation can be visualized as an activity queue (Figure 3), with the top activity being the current focus. As the agent is interrupted, the order of the activities in the queue can change. If operators are interrupted frequently, the delay in re-orienting to new activities can be increased (e.g., due to stress or memory limitation). Agents can be interrupted by various communication modalities including Chat, E-mail, telephone, and in-person, with a notional representation of the disruptiveness of each modality shown in Figure 3. Subject matter experts have provided the following guidance on how much each modality is used (in terms of an Average Utilization Rate) in a typical Command and Control environment.

- Chat: 75%
    - Text Chat: 70%
    - Audio Chat: 5%
- E-mail: 10%
- Telephone: 5%
- In-Person: 10%

The agent behavior will also depend on the agent's past experience with the source of the interruption. Certain behavior is expected, depending on whether the source is trustworthy (McCarter and White, 2007), for example.

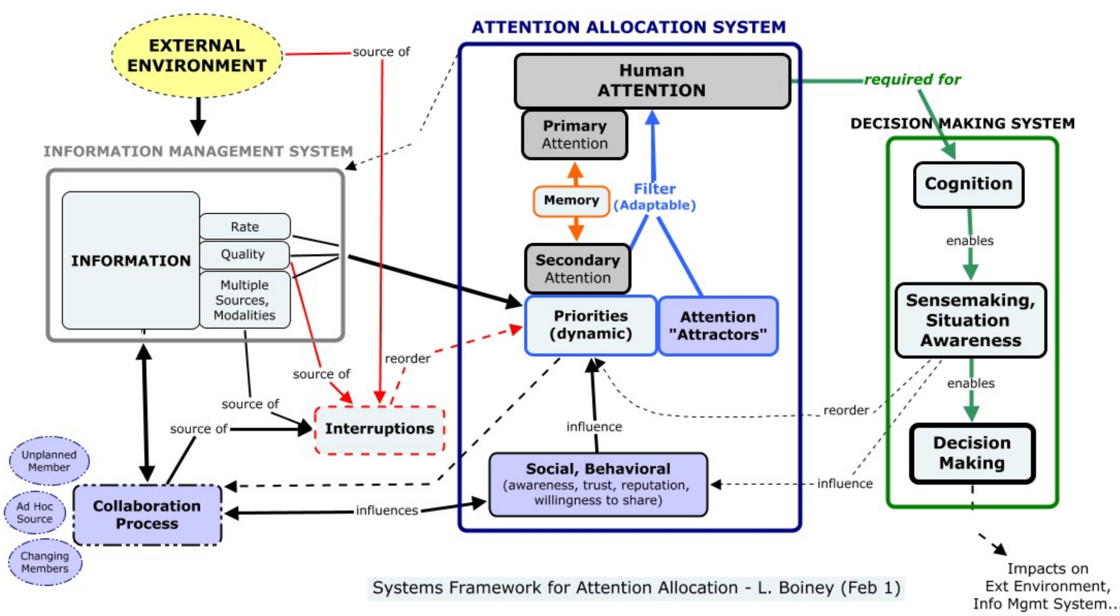

Figure 2. Systems framework for attention allocation (Boiney, 2007).

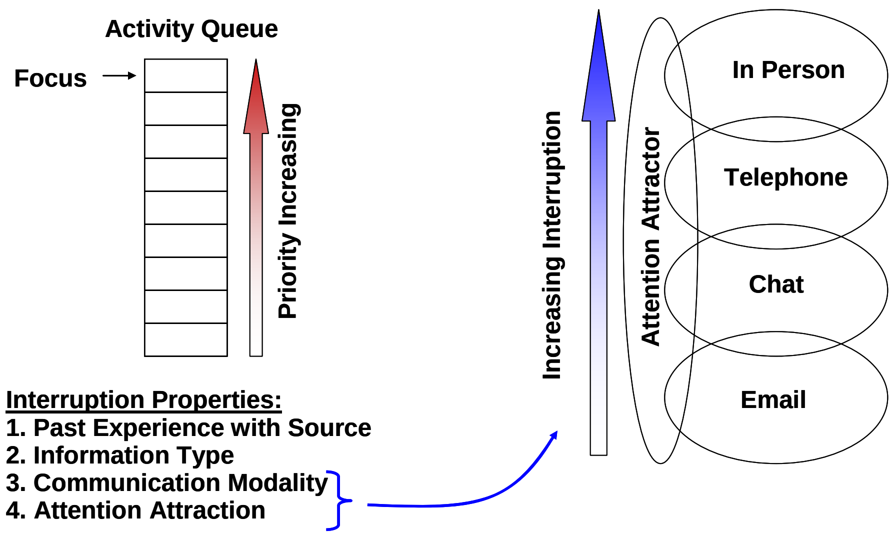

Figure 3. Operator activity queue and communication modalities.

The information itself will also affect agent behavior—it may be specifically related to the activity in which the operator is working, change the operator's priorities, or is a distraction (e.g., media report or pilot-down). Therefore, each interruption is associated with: 1) the agent's past experience with the source of the information; 2) the information type; 3) a communication modality; and 4) whether or not an attention allocation attractor is present. Attention attractors are defined as elements of higher inherent interest, including being drawn to human faces and voices (especially those indicating emotion), changes in the environment (if not too gradual), informal conversation (versus formal language), bright colors, signs of imminent danger, and anything novel or unexpected or unresolved (Boiney, 2007). For example, operators sometimes scan 10 Chat rooms when only 4 are actively used. In this instance, one question is whether it possible to develop a smaller interface (e.g., summary display) that still provides value to the operator? Such a display can also be designed to provide greater value to the operator through the use of social network software and rule creation (Boiney, 2007). The degree of disruptiveness of the attention attractor can span a wide range as illustrated in Figure 4.

The scenario selected for the operator-environment model was the Time Sensitive Targeting (TST) process, which is modeled as 49 operators working in the processes outlined in Figure 4. These operators work in a specific group denoted by the red boxes. In the current version of the model, one group of 8 operators interact to complete a task (rectangles in Figure 2), based on a random number of activities or sub-tasks. Each agent is randomly assigned a number of static properties, including a level of expertise, experience, and ease of interaction.

Based on the initial properties, the agent develops a social network as the simulation is run. As shown in Figure 5, an agent "grabs" a critical event. The agent checks to see if the event needs any more information to be handled properly. If not the agent is done, and the processing proceeds depending on a random setting of the modality of the next interaction. If the modality is "Talking" a check is made to see if the given agent has any trusted associates from the same cell. If so the agent picks one and talks for a random amount of time. If the modality is "telephone" no associate is picked but a "talk-time" is calculated (assumed outside of cell). In either case, after the talk-time passes the agent checks if success in getting information was accomplished. If the agent is successful (a probabilistic outcome), the amount of information needed to handle an event is decreased, adds the associate agent to the trusted list (if not

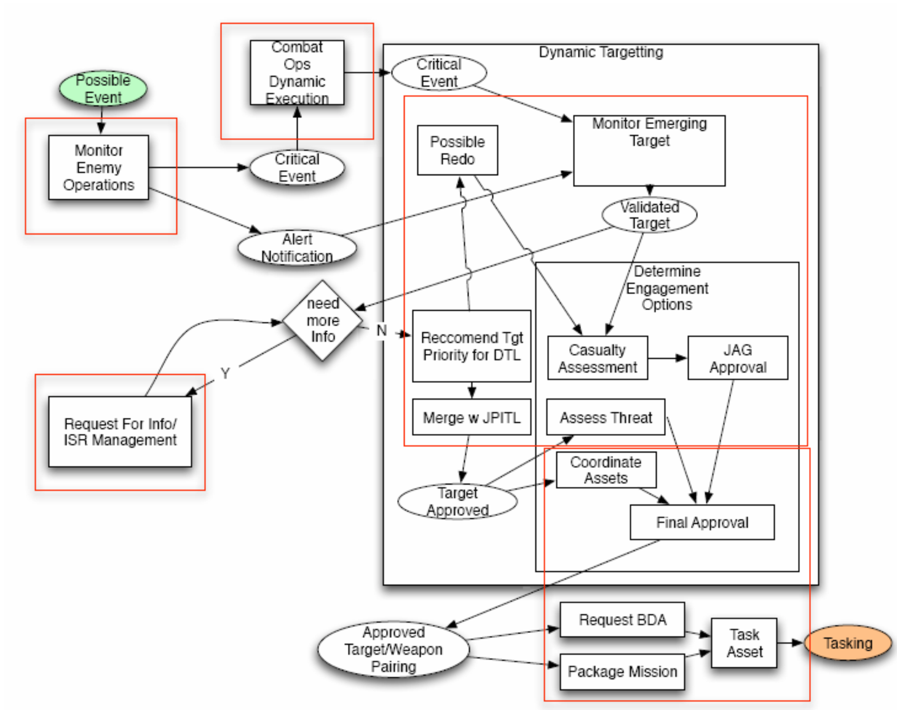

Figure 4. Air and Space Operations Center (AOC) process model. The rectangles indicate tasks and arrows indicate the flow of critical events—Dynamic Target List (DTL), Joint Integrated Prioritized Target List (JIPTL), Judge Advocate General (JAG), and Battle Damage Assessment (BDA).

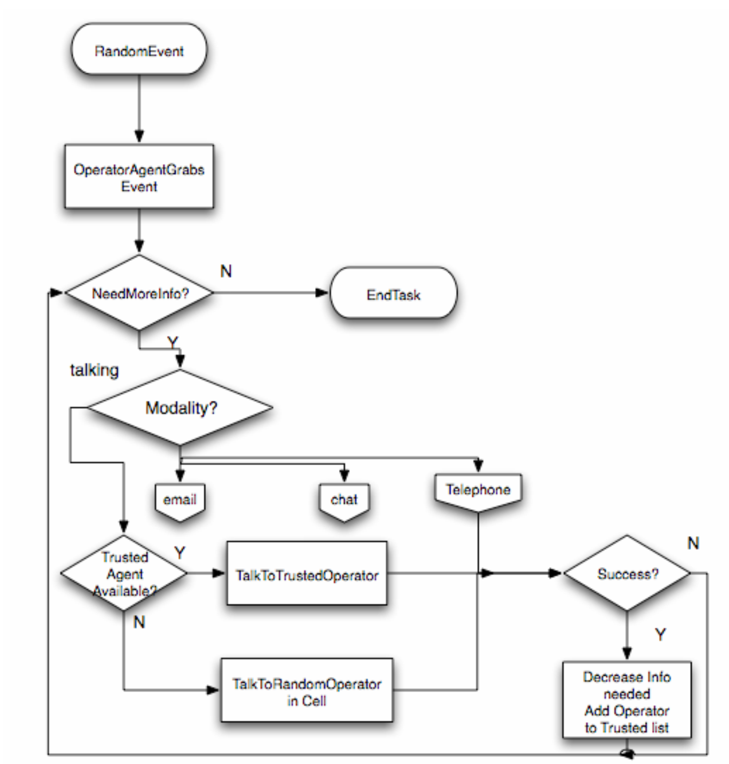

Figure 5. Modeling operator activities or sub-tasks in the Agent-Based Model.

already there), and goes back to see if the event needs any more information. If the agent was unsuccessful, a check is made to see whether more information is needed. E-mail and Chat modalities are implemented in a simplified fashion.

Agent Measures of Performance (MOP) or indicators include: transition time between activities or sub-tasks, errors, missed information, response time, and repeated requests. Potential experiments within the Agent-Based Modeling environment include:

- Add AOC environment distractions (e.g., media reports or pilot-down)
- Use of attention attractors (e.g., summary display)
- Incentives that motivate operators to collaborate between cells
- Remove well-contacted operators (i.e., how important are personal relationships?)
- Create environment that has more remote cell members (e.g., distributed AOC)
- Create procedures that increase the chance of bottlenecks (e.g., require all information to pass through one agent)

A pilot-down scenario was developed based on the observation that many AOC operators can become distracted by a pilot-down even if the operator is not able to directly assist in the rescue. Furthermore, this distraction has been hypothesized to have a detrimental effect on the activities being handled by the non-involved operator. Figure 5 shows the agent environment (Java main class) and the pilot-down event timer as well as the timer for other TST events. Circles indicate global variables and squares indicate Java classes (e.g., state charts for operators and Chat are implemented). Figure 6 depicts the agent environment, Figure 7 shows the operator class in detail, and Figure 8 provides state charts for operator activity and availability.

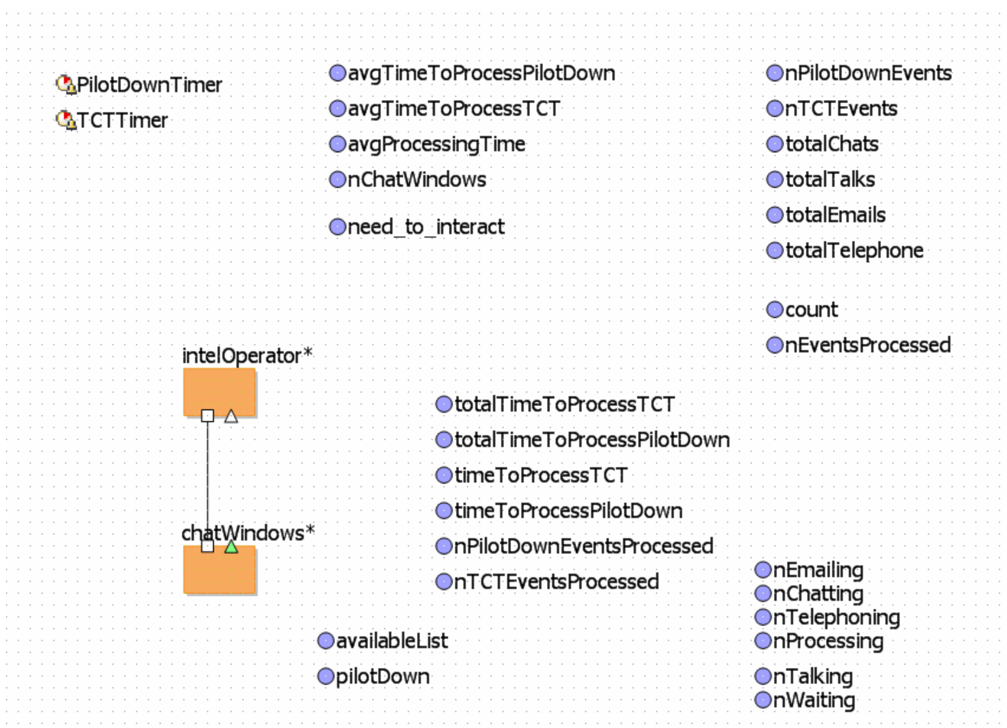

Figure 6. The agent environment or Java main class.

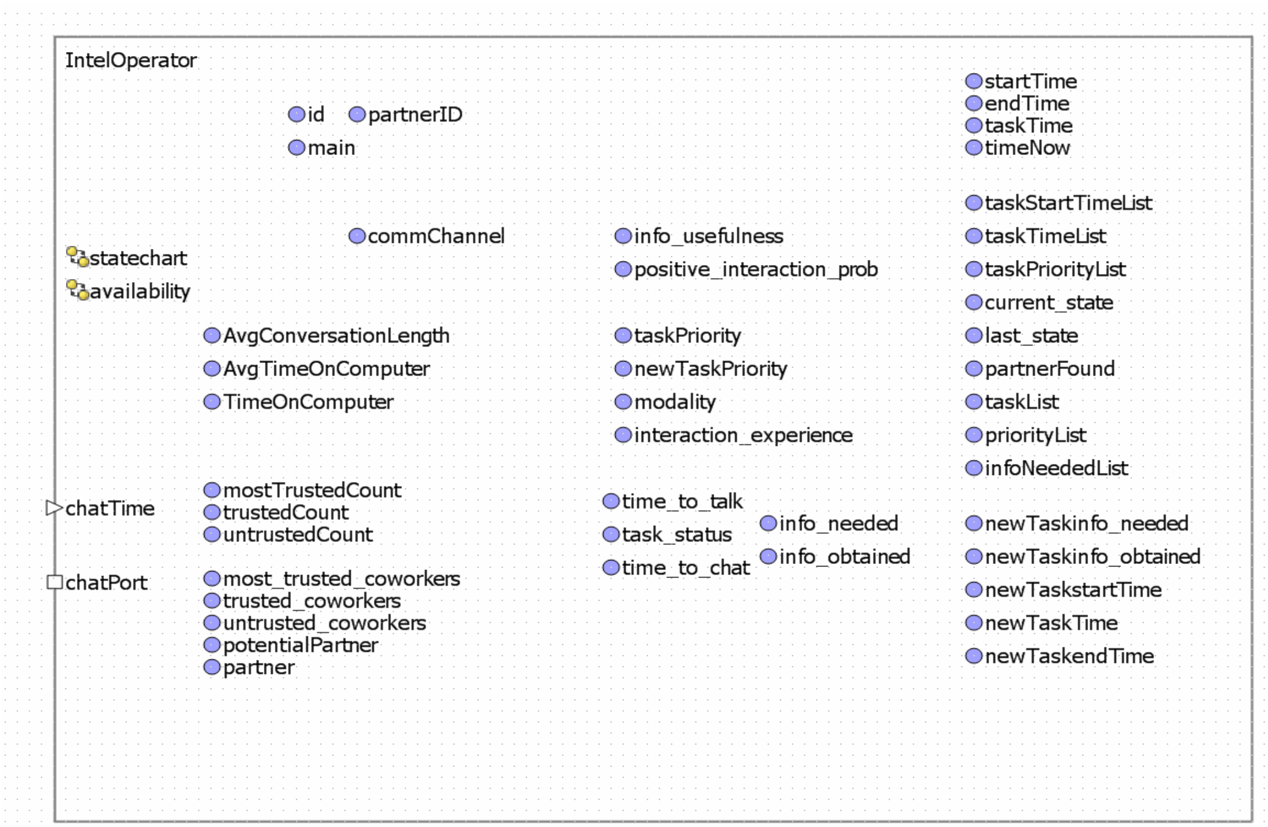

Figure 7. The operator class.

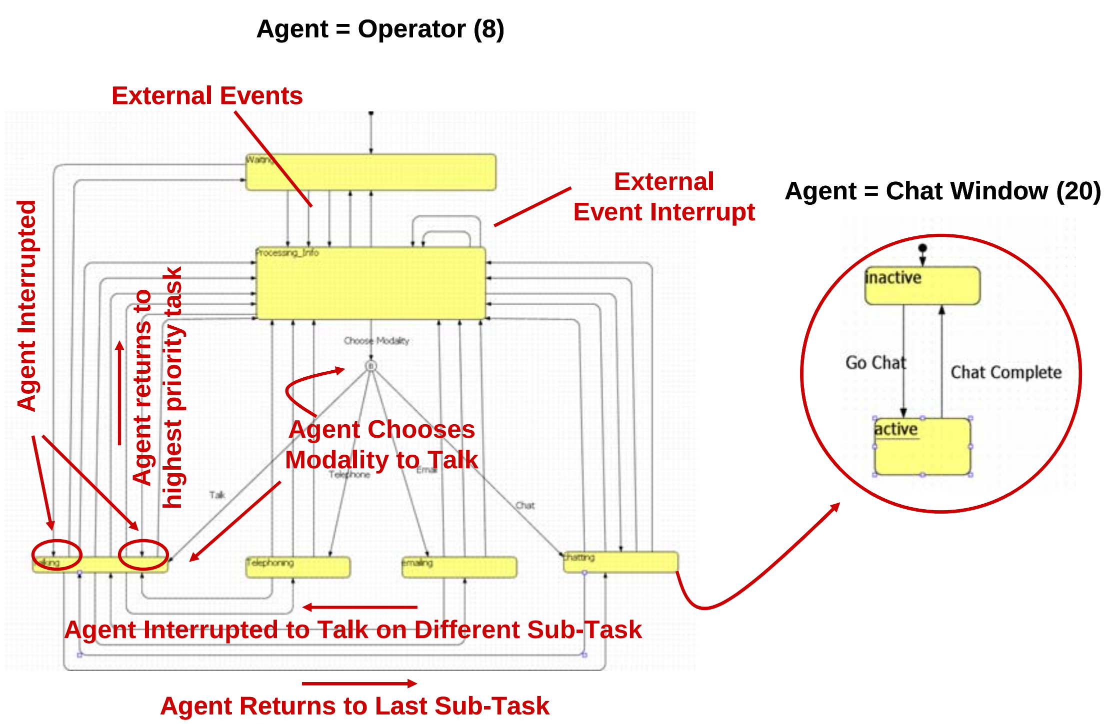

Figure 8. The operator state charts for activities (left) and chat window state chart (right).

AOC Process Model (Petri Net)

The existing Air and Space Operation Center (AOC) process model (Figure 3), describes Situation Awareness and Assessment (SA&A) and Time Sensitive Targeting (TST) operations, including human resources (Mathieu *et al*., 2007). The strike and support missions included six different types of critical events: 1) Theater Ballistic Missile (TBM) launch; 2) TBM Detection; 3) Combat Search and Rescue (CSAR, a pilot-down situation); 4) Surface to Air Missile (SAM) radar emissions; 5) Choke Point (enemy assets constrained by terrain); and 6) Air Tasking Order (ATO) re-tasking.

All events were given the same priority with the exception of a CSAR event, which was given a higher priority. The effect of this was that the higher priority event preempted work on any other event type being processed in the same resource. All events trigger TST responses by the Dynamic Targeting Cell (DTC) of the AOC in addition to their responsibilities of monitoring the progress of the ATO. The focus of this study is on DTC operations during the execution of the campaign as targets of opportunity become available. The process of time-sensitive targeting is outlined in Figure 4.

The AOC process model built using MSim (Mathieu *et al*., 2007) explores and optimizes the operational processes of the DTC—Figure 9 shows a portion of the model. The Petri net model is used to find efficient results for various indicators of performance regarding TST performed in the DTC, including: 1) Time from target appearance to target prosecution—critical event response time; 2) Workload in DTC—resource utilization; and 3) Number of operators in DTC. However, making the DTC as efficient as possible without considering global environment factors may lead to a problem where the “local” optimum produces a result at the system scale that is below the “global” optimum.

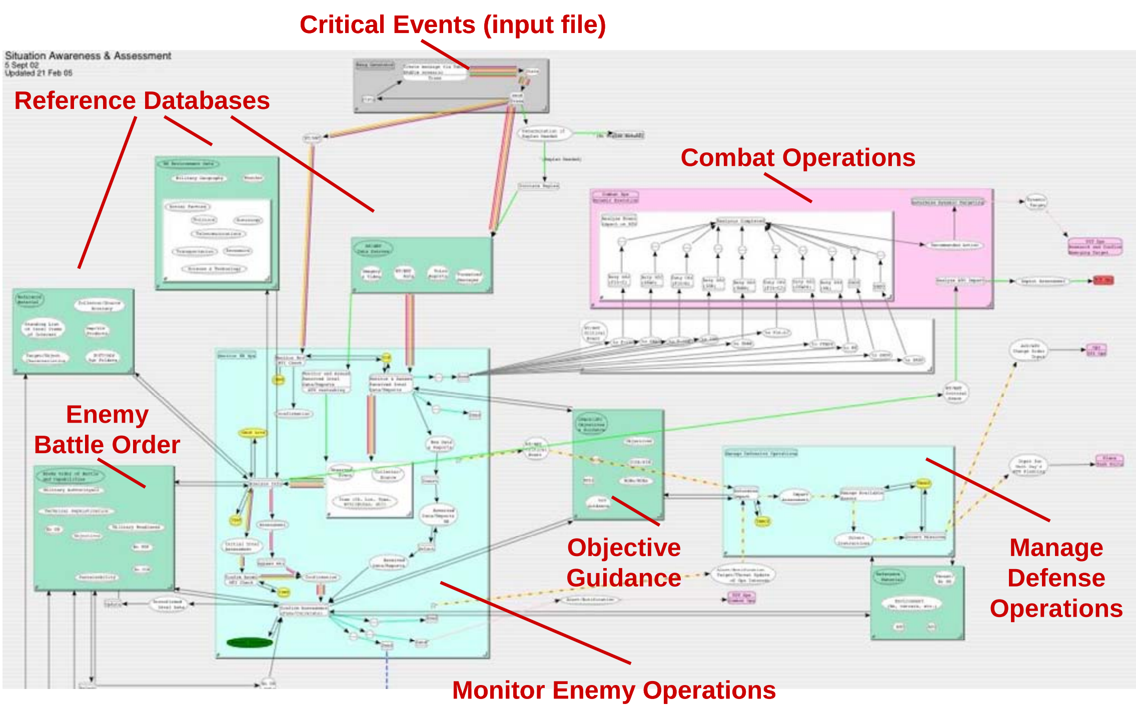

Figure 9. Portion of AOC Petri net model—Monitor Operations and Combat Operations.

Global & Mission Models (System Dynamics)

The System Dynamics modeling process was used in two distinct manners to support modeling of DTC operations: 1) Informing the AOC Petri net model of global-scale dynamics; and 2) Linking the Petri net and System Dynamics models to achieve strategic mission and global-scale simulation.

The System Dynamics model is composed of 4 sections (Mathieu *et al*., 2007). Input values critical event response time (Figure 10) and personnel utilization from the AOC process model are introduced, and behavior derived from the input values are modeled (Figure 11). The System Dynamics model can be simulated either by itself, or in conjunction with the Petri net AOC process model. When simulated by itself, a goal-seeking loop drives the system towards a reduction in adversary state until the Joint Forces Commander's (JFC's) goals are achieved (Figure 10) and all TBM observations cease. The adversary system produces its own responses by setting up and launching TBMs.

The Joint Forces goal-seeking process can be halted if something causes *"US political will"* to become sufficiently low (Figure 11). When simulated with the Petri net model, the goal-seeking loop is turned off, and information instead flows to and from the AOC process model.

Quantitative information can enter the System Dynamics model from the Petri net through two variables: 1) *average response time* (Figure 10) and 2) *maximum personnel utilization*, which is used to determine the probability of major errors in prosecution. Figure shows the response times for each event over the course of the day. This is the output of the AOC Petri net model at the end of day 1—average daily values for each event type are used to calibrate day 1 of System Dynamics model. The output from the System Dynamics model is a set of critical events for the next day. Therefore, the two models are run in succession to simulate the effect of global dynamics on AOC operation. Some global processes may be important in influencing mission effectiveness (completing military operations to the JFC objectives). In this case, *"world support"* for US operations, as well as *"US public support,"* both impact *"US political will,"* which decrease the degree of military actions to zero (see Figure 11).

Results and Discussion

Figure 12 shows the average event response time (y axis) for operators handling the pilot-down tasks (green) and operators handling other TST tasks (blue) for a given number of critical events. These results reflect that the pilot-down gets the top priority, thus showing a faster average response time for a given number of events.

Figure 13 shows the response times from running the AOC model for the original data file of critical events—the baseline run. More critical events in a given period, results in a longer AOC response time to accomplish the tasks (e.g., see the period between 10 and 13 hours). This indicates that operators are increasingly busy, and there is a queuing of tasks. As events become less frequent, the queues are reduced and the response time reduces (e.g., period after 13 hours). A weighted average response time was calculated for the TBM launch and detect events. This value was used in the System Dynamics model—Figure 10 *"average response time."*

The highest utilized operators in the AOC model were determined using the baseline run and are shown in Table 1. This maximum personnel utilization rate (e.g., 65%, Coordinate Airspace) was used in the System Dynamics model.

Using the critical events in the baseline run, the AOC model was executed and data was passed to the System Dynamics. The System Dynamics model was run for 1 day and the output of critical events was then used to run the AOC model again. This process was done 9 times and the System Dynamics results are shown in Figure 14. The *"adversary military capability"* starts at 100% and is drawn down over the 9 days. The TBM setup and launches first increase and then decrease as the capability is drawn down. U.S.

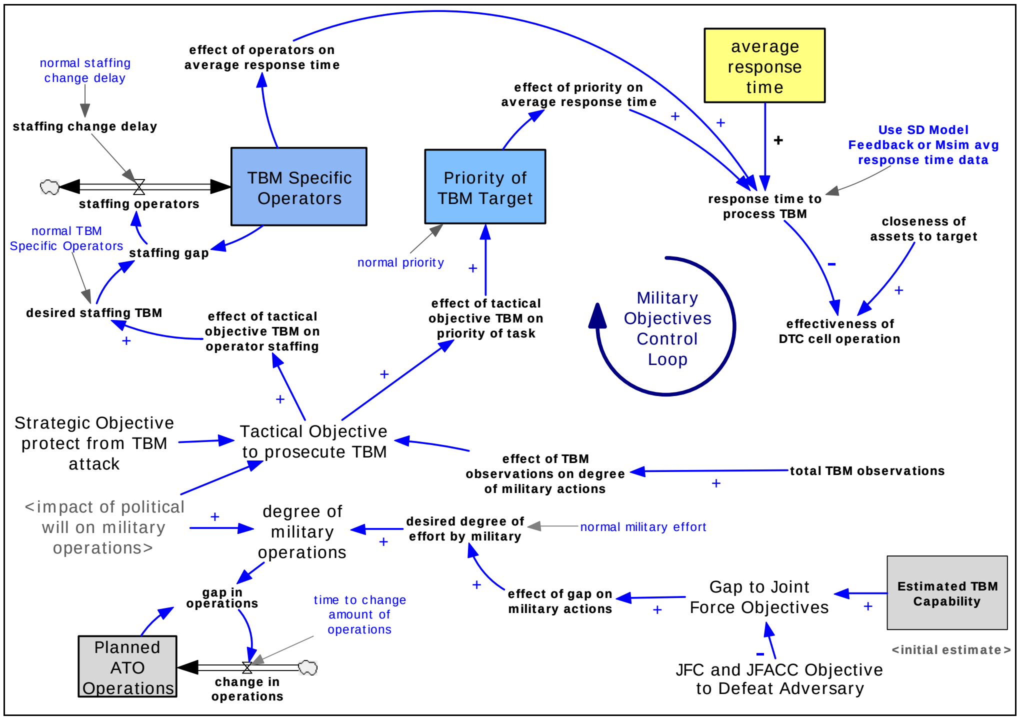

Figure 10. System Dynamics model of DTC operations for TST of adversary TBMs.

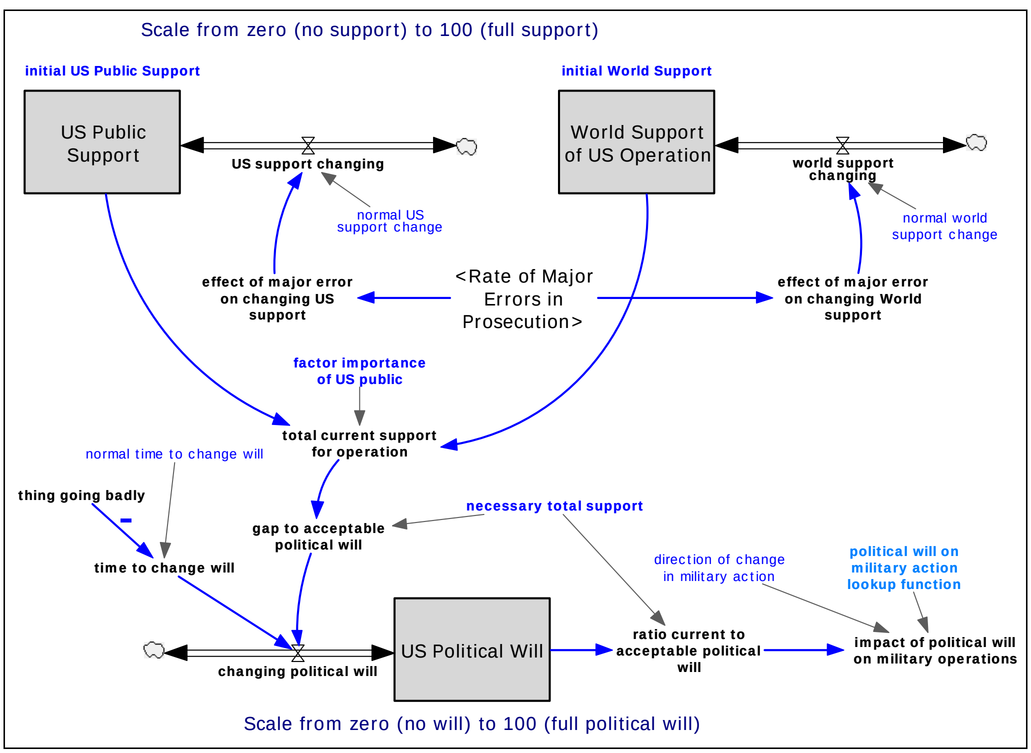

Figure 11. US and world public support and its effect on political will.

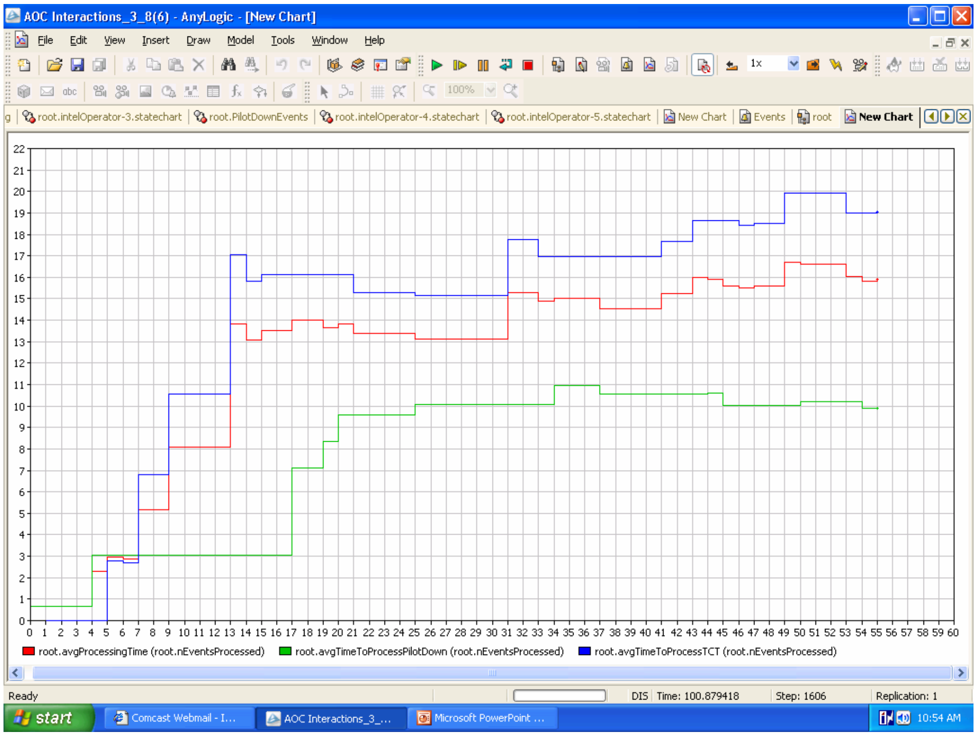

Figure 12. Results for average response time vs. number of critical events from the Agent-Based Model for a random number of activities or sub-tasks for the pilot-down operators (green), the Time Sensitive Targeting operators (blue), and the average of both (red).

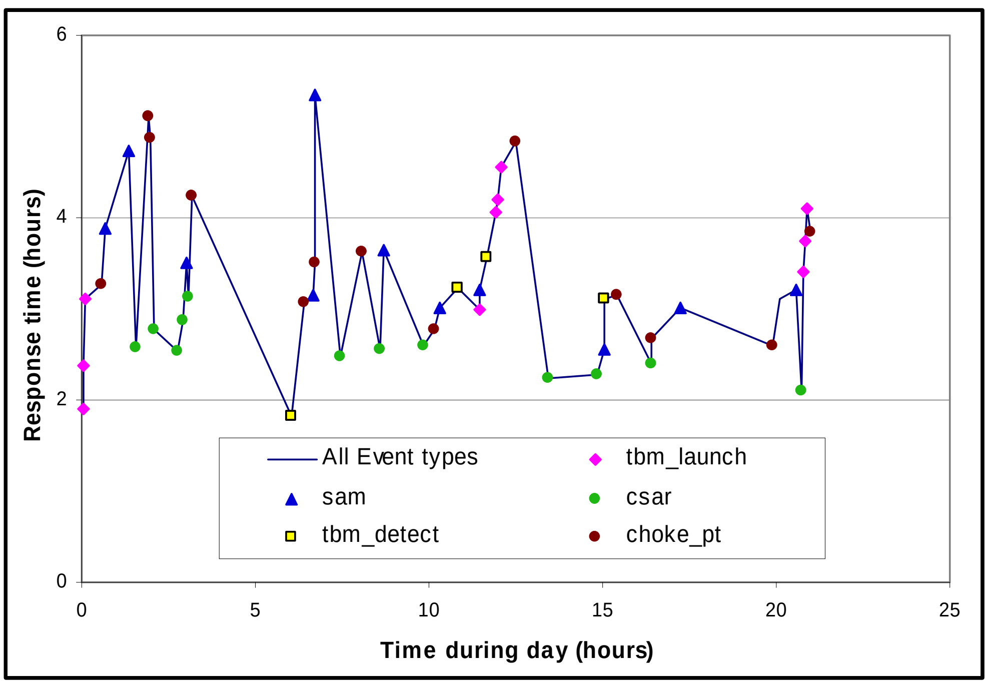

Figure 13. The response time for each type of event over the course of one day (baseline run). CSAR or pilot down has the highest priority and the remaining 4 events have the same lower priority.

Table 1. Highest utilized AOC operator from Petri net (day 1).

Utilization Rate	**AOC Cell**	**Job Description**
65 %	*Coordinate Airspace*	Coordinate airspace
62 %	*Combat Operations* *Evaluate Current Assets* *Package Mission* *Get Approvals*	Analyze event impact on ATO
61 %	*Combat Operations* *Dynamic Targeting*	Determine impact; Research and monitor
60 %	*Combat Operations* *Dynamic Targeting*	Assess threat environment; Collect intelligence; Monitor emerging targets (top secret)
57 %	*Monitor Operations* *Dynamic Targeting*	Determine impact; Research and intelligence
55 %	*Combat Operations* *Evaluate Current Assets* *Package Mission* *Get Approvals*	Asset availability; Approve package for mission; Analyze event impact on ATO; Manage assets

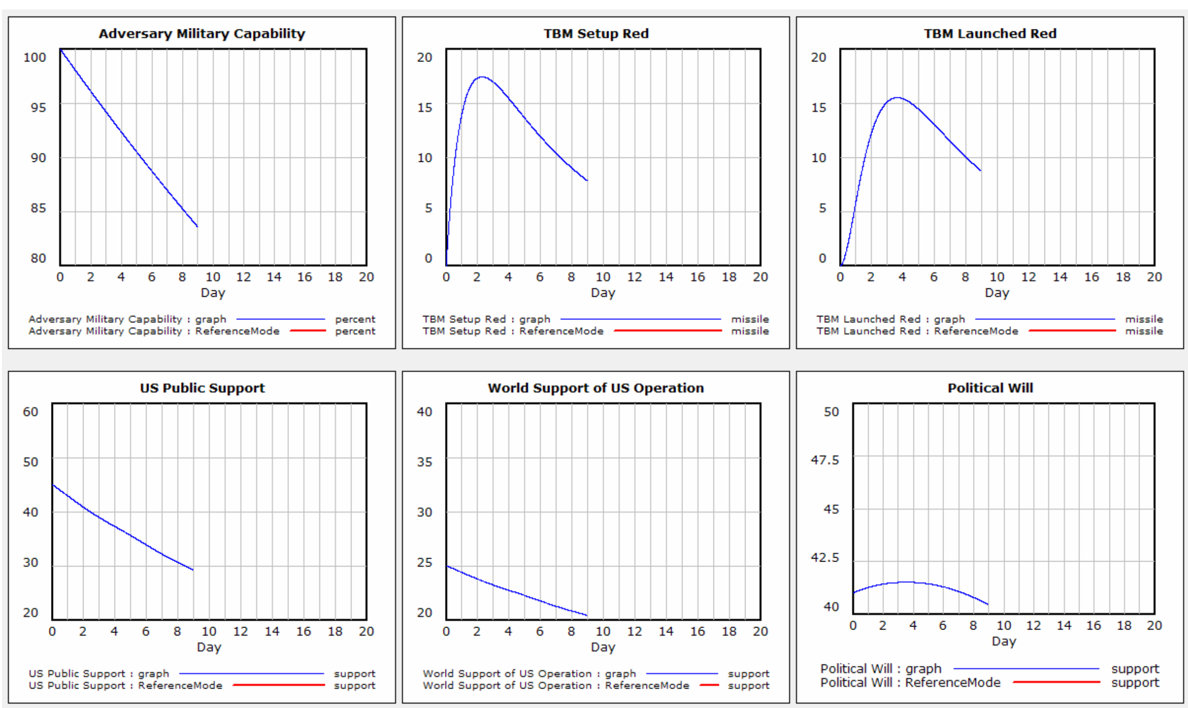

Figure 14. The System Dynamics model output for 9 days.

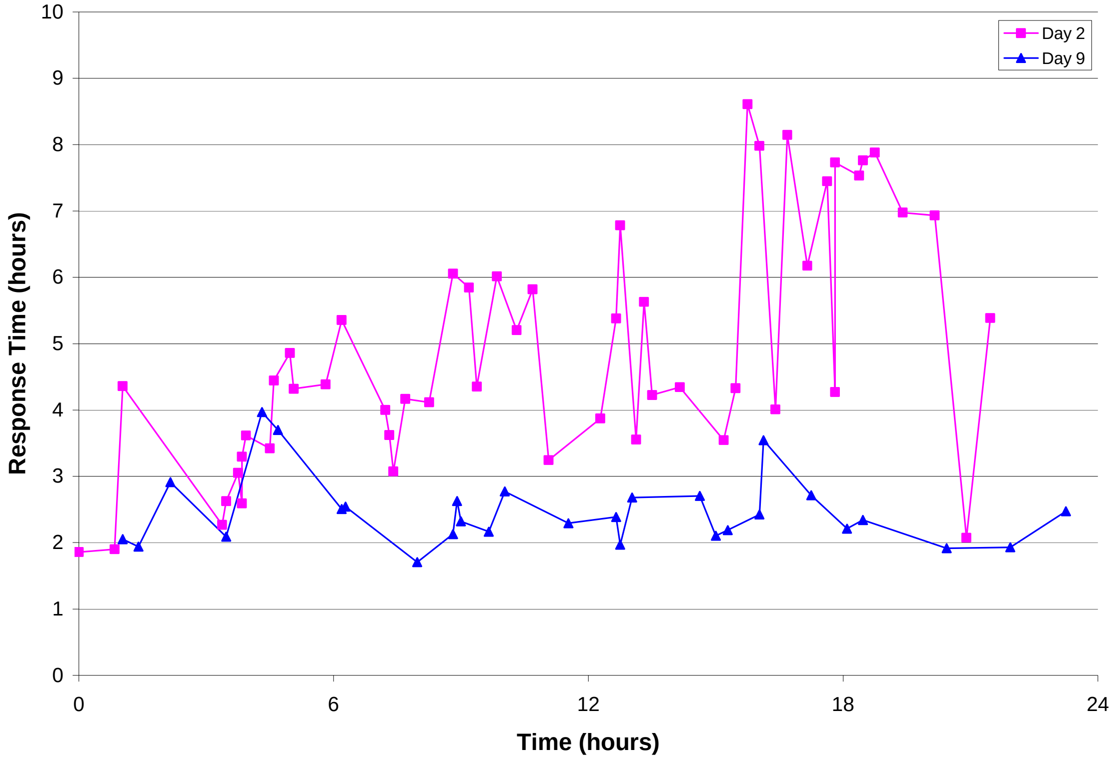

Figure 15. The response time for all events for the 2nd and 9th days.

and world public support both decrease and political will first increases slightly and then decreases. Figure 9 shows the critical event response time from the Petri net model for day 2 and day 9. At the beginning of the campaign, there are more critical events and the AOC model shows that there is queuing with a maximum response time of 8.5 hours. As the critical events are reduced by taking out *"adversary military capability,"* the AOC model shows that queuing is reduced and response time is consistently between 2 and 4 hours. These results show that the System Dynamics model has been calibrated to the Petri net model. This is the first step in model validation.

Conclusions

A preliminary Agent-Based Model has been implemented that accurately reflects event priority—pilot-down events are handled faster than other Time Sensitive Targeting events. Next, the Petri net model will be evaluated for use in determining information overflow indicators for operators (e.g., operator stress)—implementing the attention allocation model is the first step. Operator overflow inherently have components that depend on individual characteristics. In order to gain insight into these issues, it is necessary to model individual scale dynamics. Modeling the AOC at the operational process scale and operator interactions at the individual scale is hypothesized to provide insight into operator overflow. The models and insight gained can be used by operators, decision makers, and new team members to understand the effects of polices that improve operations on the overall system.

One of the agent Measures of Performance or indicators is errors. Future work will include modeling errors as a function of the number of activities or sub-tasks, the difficulty level of the sub-task, and the number of interruptions multiplied by the duration. This may be an example of the Agent-Based Model having direct feedback to the System Dynamics model as shown in Figure 1.

The critical events processed started at 50 on day 1 (Table 2) and were drawn down to 29 on day 9 in the integrated Petri net and System Dynamics models. The corresponding response time and maximum personnel utilization are also shown in Table 2, and are reduced as expected. This drawdown scenario was selected for the first step of validation. Future scenarios could include:

- Have *"adversary military capability"* rise due to the purchase of weapons resulting in more critical events to handle. This would cause utilization to increase causing more errors in prosecution and possible effect on U.S. and world opinion.

- Remove a particularly well-connected operator from the AOC resulting in more utilization and higher response times.

- Currently, the CSAR or pilot-down is the highest priority event, but TBM events targeting a densely populated area could become the highest priority as dictated by the global model. This would result in faster response times for TBM, but may cause negative U.S. opinion.

The AOC process model is, however, linked (e.g., critical event process time and probability of errors) to a global-environment model that is driven by the political landscape in which the AOC operates. Future work in linking the Petri net and System Dynamics models include feedback from the global model's *"priority of target"* and *"operators"* available to the AOC—these variables will reflect changing *"political will"* and *"JFC objectives"*.

Table 2. The average response time and maximum personnel utilization for each day with corresponding number of critical events.

Day	Response Time (hours)	Maximum Personnel Utilization (%)	Number of Critical Events
1	4.33	64.5	50
2	5.40	66.1	51
3	3.40	62.0	45
4	3.54	53.9	39
5	2.79	48.8	37
6	2.76	52.3	37
7	3.09	54.5	38
8	2.34	50.3	31
9	2.37	42.7	29

References

Boiney, L.G. 2007. More than information overload: Supporting human attention allocation. 12th International Command and Control Research and Technology Symposium (CCRTS).

Kuras, M. L., and B. E. White. 2005. Engineering Enterprises using Complex-System Engineering. INCOSE 2005 Symposium, 10-15 July 2005, Rochester, NY.

Kuras, M. L., and B. E. White. 2006. Complex Systems Engineering – Position Paper: A Regimen for CSE. Fourth Annual Conference on Systems Engineering Research (CSER), 7-8 April 2006, Los Angeles, CA.

Mathieu, J., Melhuish, J., James, J., Mahoney, P., Boiney, L., and White, B. 2007. 11-12 January 2007. Multi-scale modeling of the Air and Space Operations Center. Symposium on Complex Systems Engineering, The Rand Corporation. `http://cs.calstatela.edu/wiki/index.php/Symposium_on_Complex_Systems_Engineering`

McCarter, B.G., and White, B.E. 2007. 15 March 2007. Collaboration/Cooperation in sharing and utilizing net-centric information. Conference on Systems Engineering Research (CSER), 14-16 March 2007, Stevens Institute of Technology, Hoboken, NJ.

White, B. E. 2005. Engineering Enterprises using Complex-System Engineering (CSE). Presentation to 1st Annual System of Systems (SoS) Engineering Conference, 13-14 June 2005, Johnstown, PA

White, B. E. 2005a. A Complementary Approach to Enterprise Systems Engineering. National Defense Industrial Association, 8th Annual Systems Engineering Conference, San Diego, CA, October 24-27, 2005