Adversarial intent modeling using embedded simulation

James Melhuish, Nicholas J. Pioch, Andrew Seidel

BAE Systems Advanced Information Technologies
6 New England Executive Park, Burlington, MA, USA 01803
james.melhuish@baesystems.com

Abstract

To foster shared battlespace awareness among air strategy planners, BAE Systems has developed Commander's Model Integration and Simulation Toolkit (CMIST), an Integrated Development Environment for authoring, integration, validation, and debugging of models relating multiple domains, including political, military, social, economic and information. CMIST provides a unified graphical user interface for such systems of systems modeling, spanning several disparate modeling paradigms.

Here, we briefly review the CMIST architecture and then compare modeling results using two approaches to intent modeling. The first uses reactive agents with simplified behavior models that apply rule-based triggers to initiate actions based solely on observations of the external world at the current time in the simulation. The second method models proactive agents running an embedded CMIST simulation representing their projection of how events may unfold in the future in order to take early preventative action.

Keywords: Adversary modeling, intent inference, model integration, agent-based modeling, system dynamics

1. INTRODUCTION

Recent military operations have demonstrated an urgent need for the U.S. Air Force and the joint community to expand the scope of its intelligence, planning, and operations beyond traditional force-on-force strategy against enemy military and infrastructure targets to include political, economic, social, and information considerations as well. Previous attempts in the modeling and simulation community to model such non-kinetic effects have been limited to non-executable conceptual models that are designed to describe but not actually simulate dynamic, complex interactions (Biggie 2003). Other approaches that seek to model non-kinetic factors as scalar force limiters or multipliers within existing large-scale executable legacy simulation environments offer only limited insight into non-kinetic and kinetic interactions due to lack of explicit non-kinetic process models and actual feedback to the kinetic models (Bross 2005). To address this problem, under the AFRL Commander's Predictive Environment program, BAE Systems Advanced Information Technologies developed the Commander's Model Integration and Simulation Toolkit (CMIST) as an integrated exploratory modeling environment combining multiple simulation paradigms such as system dynamics, Bayesian cause-effect models, and agent-based discrete event models, to enable rapid forecasting of battlespace effects (see Figure 1). For technical details on CMIST, see Pioch 2007 and Pioch et al. 2007.

Through the first two years of development, we have applied CMIST's hybrid modeling environment to develop strategic and operational models of varying scale for notional scenarios, including a small-scale counter-insurgency vignette and a larger-scale air- and ground-based combat operation in which CMIST was used to compare effects across alternative courses of action with vs. without initial PSYOPS missions (Pioch et al. 2008). Both of these modeling efforts used a simplified approach to agent intent modeling in which an agent compares one or more observed properties of the outside world to specified thresholds and initiates particular actions by changing one or more variables in the world model under its control. We refer to this as *reactive intent modeling* in the sense that the agents are simply responding to world events as they occur. Recently, we have extended CMIST to allow an agent to run an embedded CMIST simulation model representing its own internal, possibly simplified, model of the outside world. This *proactive intent model* allows the agent to project the future state of the world, including adversary actions, in order to take appropriate preventative measures. We discuss in Sections 3 and 4 comparative results using reactive vs. proactive intent models in the context of our notional PSYOPS-Combat scenario. We preface these discussions with a brief overview of CMIST's system architecture in Section 2.

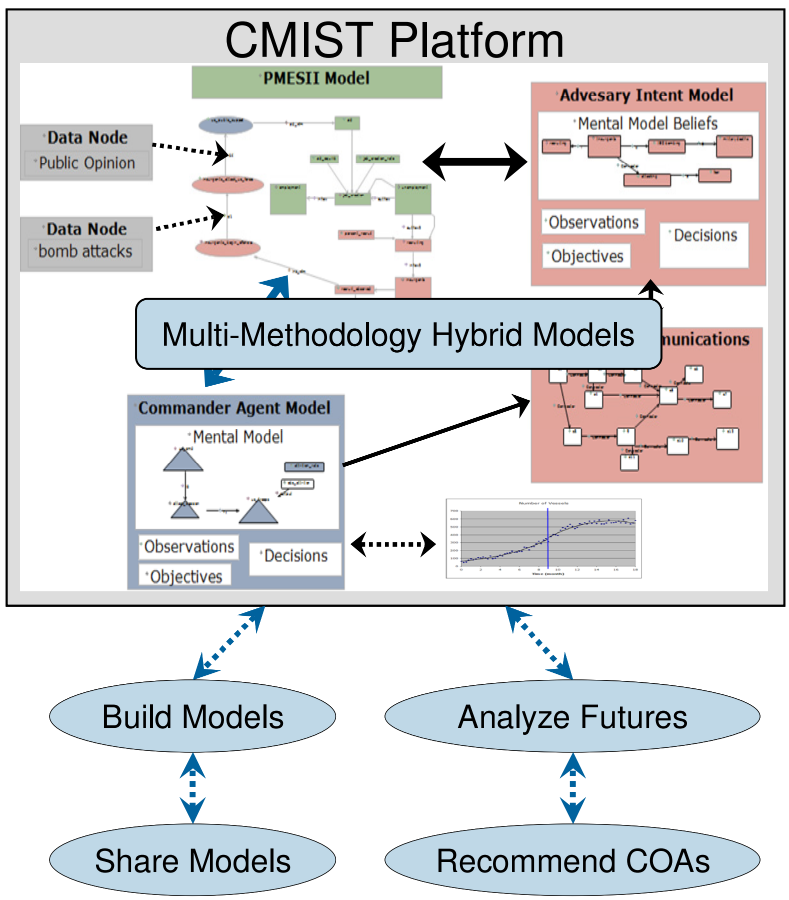


Figure 1. CMIST's hybrid modeling environment was extended to support advanced intent modeling via embedded simulation and Bayesian Knowledge Bases for inference of goals and beliefs.

2. System Architecture

CMIST is an Integrated Development Environment (IDE) for model authoring, integration, validation, and visualization. It is built on the versatile Eclipse framework, a widely used open source Java IDE. CMIST utilizes recent advances in model-driven architecture (MDA) to automate the process for integrating new modeling paradigms. The MDA process in turn is based on proven software engineering standards for model representation and exchange, such as Unified Modeling Language (UML) and eXtensible Markup Language (XML). CMIST is designed with a two-stage architecture for distinct categories of users (see Figure 2):

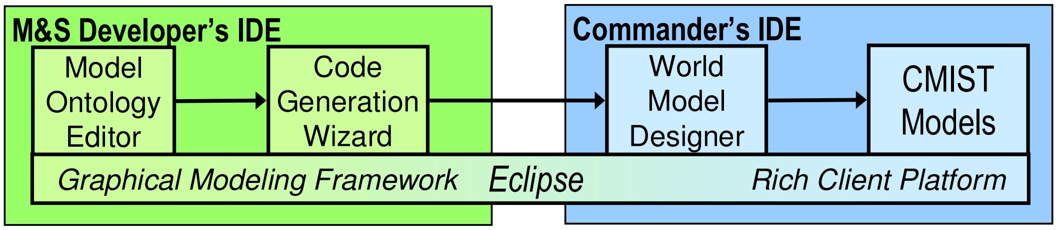


Figure 2. CMIST's two-tier architecture with IDEs for Modelers and for Developers

1. A Developer's IDE for simulation and software developers to rapidly incorporate new simulation methodologies and tools, making them available for use in the Commander's IDE. The Developer's IDE provides the shared representation and common repository for model description and methodology characterization, such as timing, and method of computation. Eclipse's Graphical Modeling Framework (GMF) is used to design a representation for the graphical modeling primitives of a new methodology and automatically generate the Java source code to implement the data classes, controller, and editor palette for the methodology. After code generation, one need only implement a *mediator* to execute a specified increment of simulation time in a native simulation tool for the chosen methodology and to translate from that tool's native data model to CMIST's shared representation.
2. A Commander's IDE for commander and supporting staff to author models, integrate hybrid models via transform links, execute models over a specified time horizon, and debug the integrated models via intuitive visualization displays, including time series charts and map-based overlays.

CMIST includes an extensible library for creating reusable data transforms that enable exchange of data between different modeling families. CMIST also provides multiple interaction patterns to synchronize multiple native simulations with disparate modeling paradigms, such as discrete time/event simulation, continuous-time simulation, and iterative solution methods such as Monte Carlo sampling.

Prior to running a model, CMIST uses the transform links to partition the overall model into fragments that each rely only on a particular methodology, as shown in Figure 3. It then compiles each fragment into a corresponding *native model* in an underlying third-party tool responsible for execution of that methodology. Finally, the CMIST simulation engine executes the model according to the chosen interaction pattern, for example, stepping time forward in each native model and then flowing outputs across transforms to the inputs of the next fragment.

The initial CMIST release successfully integrated three modeling methodologies and native tools, as shown in Figure 3:

- Probabilistic cause-effect modeling enabled by dynamic Bayesian network algorithms from AFRL's Operational Assessment Tool (OAT) (Gossink and Lemmer, 2004)
- System Dynamics modeling based on (Forrester 1961), enabled by U.C. Berkeley's open source Ptolemy II framework (Eker et al. 2003)
- Agent-based modeling enabled by Telecom Italia Lab's open Java Agent Development (JADE) framework (Bellifemine et al. 2003)

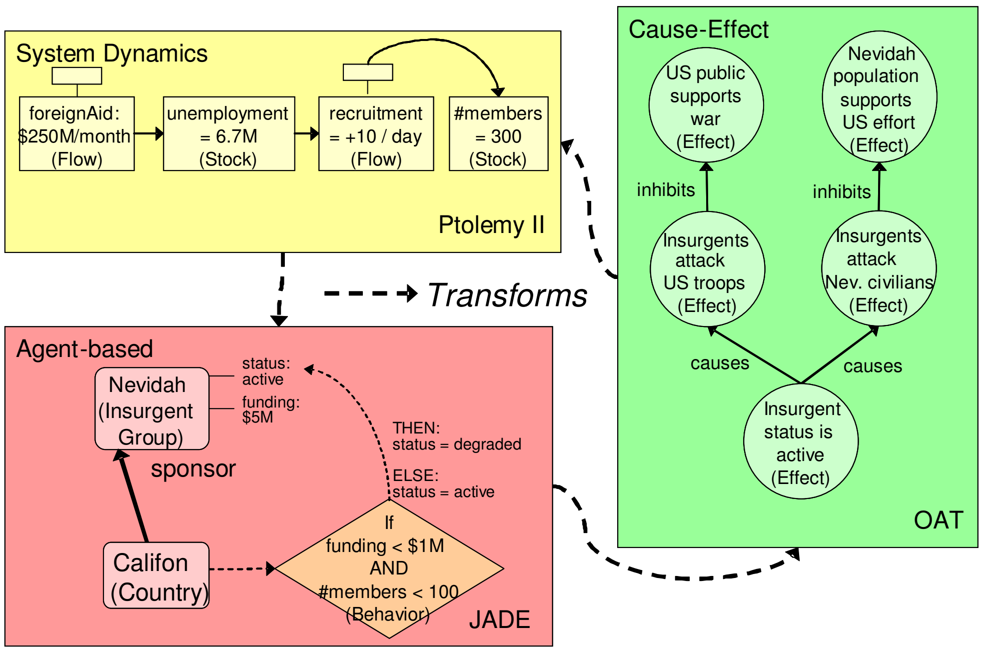


Figure 3. CMIST automatically compiles, executes, and integrates model fragments for multiple methodologies

3. Reactive Intent Modeling

This section describes a baseline Insurgent Growth model that is used as the backdrop for experimenting with reactive vs. proactive agent decision-making. The model is based on a larger-scale Military-Social-Information model investigating the effects of a non-kinetic PSYOPS campaign on major combat operations against a fictional adversary, Califon, which attempts to invade a neutral neighbor, Nevidah, in order to gain control of valuable mineral fields near the border between the two countries. For more details on this combined PSYOPS-combat model see Pioch et al. 2008.

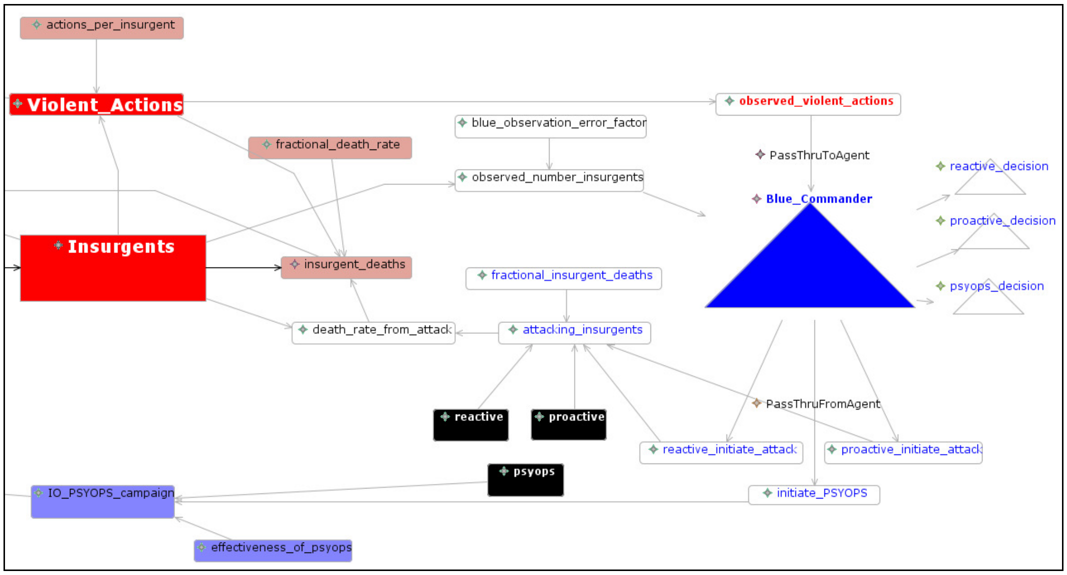
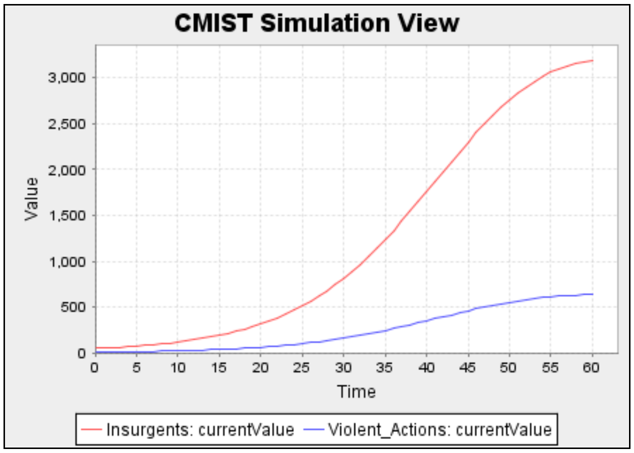


Figure 4. CMIST model fragment with switches for reactive vs. proactive counter-insurgency attack strategies.

The major dynamic of interest here is growth of **Insurgents** due to a positive feedback loop (not shown in Figure 4) that exploits fear among the general population to convert additional insurgents. This results in an S-shaped growth pattern (Figure 5, left). The **Blue_Commander** can observe the amount of violence and can observe an estimated quantity of insurgents. Based on this data, the Blue commander can reactively order an offensive attack to kill insurgents, or can proactively (preemptively) order the attack, and/or can mount a PSYOPS campaign to impact insurgent growth.

Three switches in the model (shown in black) provide rapid access to these options: Reactive, Proactive, and Psyops. When the Reactive switch is set to 1 and the other two are set to 0, the Blue commander agent bypasses its embedded model and utilizes built-in logic to initiate a counter-attack when **observed_violent_actions** exceeds a threshold of 50 incidents per day. The results are shown in Figure 5 (right). While the counter-attack eventually causes the rate of insurgent growth and violence to decrease, the peak violence of 75 is significantly higher than the desired threshold. Thus, by reacting only after the threshold was exceeded, the commander agent actually failed to adequately contain the level of violence. It is clear that a more proactive strategy is required, which we will demonstrate in the next section using embedded simulation.



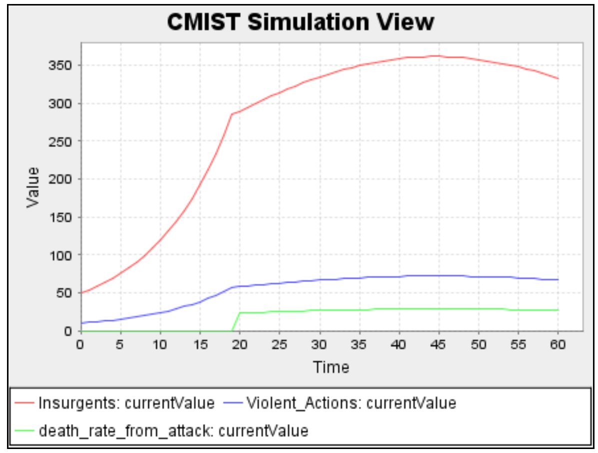


Figure 5. Insurgent growth and violence with no Blue intervention (left) vs. reactive Blue counter-attack (right).

4. Embedded Intent Modeling

4.1 Embedded Simulation Overview

In Section 3, we modeled commanders as reactive decision-makers. Here, we model commanders as proactive decision-makers. In order to be proactive, the agent must have some beliefs or projections of the *future* state of the world, rather than simply observing the *current* state of the world. To achieve this, we extended the JADE agent decision-making implementation to include a reference to a separate CMIST model. This model is embedded in the agent and is its representation of the world, a mental model of the "real" world. The *main model* refers to the primary model in CMIST that represents the real world situation (a portion is shown in Figure 4 above). This is the normal simulation model that would be built by an analyst to answer particular questions and forecast outcomes. The *embedded model* refers to a model that is built separately for use by an agent in the main model. Multiple agents may have their own model, or may share a model.

The simulation engine recursively executes embedded models. For each time-step in the main model, the embedded model simulates for a user-defined number of time steps. The simulation engine saves all output data for each individual embedded model run.

4.2 Proactive Intent Modeling

The proactive simulation allows the commander to forecast a single possible future based upon an embedded model and observations from the environment (Figure 6). The commander uses the observations to set initial parameter values in the embedded model, simulate the model for a fixed number of time steps, and retrieve the result. Here, the number of time steps is set to 20, allowing the commander to forecast 20 days into the future.

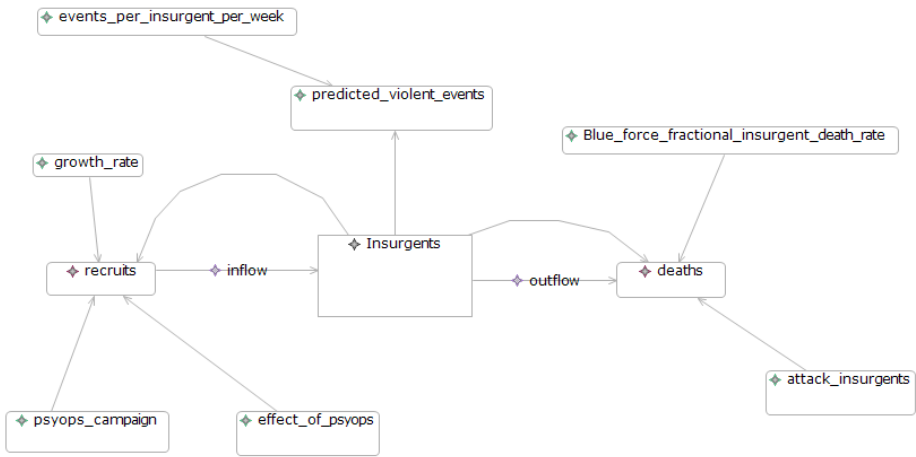


Figure 6. Commander's embedded mental model of the dynamics of insurgent growth

The core feature of the embedded model is a positive feedback loop in insurgent recruiting. When the blue **psyops_campaign** and **attack_insurgent** variables are false, the number of insurgents grows exponentially at a rate specified by the variable **growth_rate.** When **psyops_campaign** is true, the recruiting rate lessens by the scale factor **effect_of_psyops.** When **attack_insurgents** is true, the negative feedback from the SD flow node **deaths** counteracts the insurgent growth.

The commander agent's forecast from this embedded model is compared to a desired threshold value for amount of violence (a property set within the agent). When the forecast violence exceeds the threshold, the commander orders a mission to attack the insurgents.

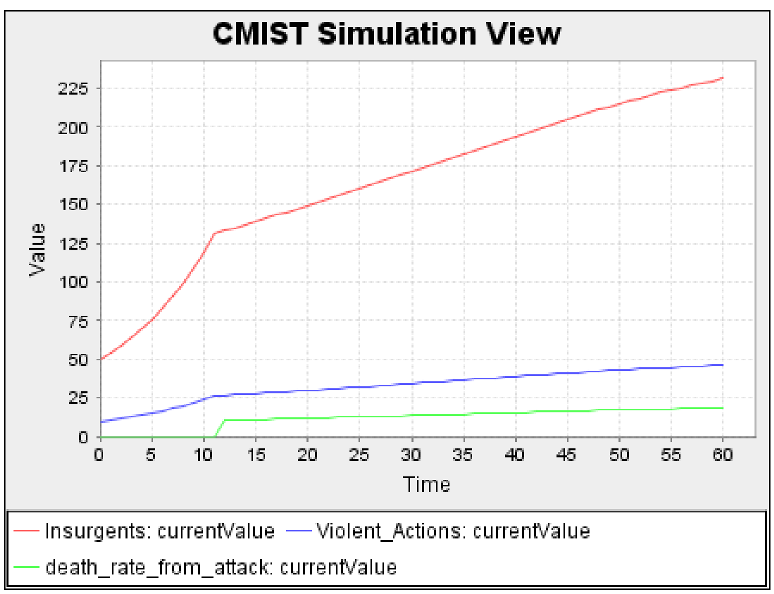


Figure 7. Proactive offensive to attack insurgents

Figure 7 shows the main model results when the proactive switch is on and the other switches are off. Note how the attack starts much earlier (Day 12 as opposed to Day 20) and leads to significantly fewer insurgents (240 instead of 360) and less violence at their peak than in the reactive scenario. Note also that the death rate of insurgents from the blue attack is smaller, because the attack commenced when there were fewer insurgents and our main model applies a fractional death rate multiplier to the number of insurgents. This proves to be a much more effective strategy than the reactive approach, since the Blue commander actually achieved its goal of keeping violence rate under the desired threshold of 50 incidents per day.

5. Conclusions

We have integrated into CMIST a new technique for adversary intent modeling that is significantly more sophisticated than the method available previously. Embedded simulation provides a means of encoding a *dynamic mental model* within an agent. Using a notional counter-insurgency model, we showed how this can be used to trigger proactive decision-making based on the agent's forecast of future state from current trends. This provides a powerful intent modeling capability for warfighters and analysts to better understand their adversaries and forecast possible futures.

Acknowledgments

This material is based upon work supported by the United States Air Force Research Laboratory under Contract No. FA8750-08-C-0101. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the United States Air Force.

References

Bellifemine F., G. Caire, A. Poggi, G. Rimassa, 2003. JADE A white paper. exp Vol 3 no 3, Telecom Italia Lab http://jade.tilab.com/

Biggie, J. 2003. *Operational Net Assessment*. Briefing to Military Operations Research Society, Decision Aids meeting, USJFCOM-J9.

Bross, P. J. 2005. *Measuring the Will to Fight in Simulation*. 73rd Military Operations Research Society Symposium, West Point, NY.

Eker, J., J. W. Janneck, E. A. Lee, J. Liu, X. Liu, J. Ludvig, S. Neuendorffer, S. Sachs, Y. Xiong. 2003. *Taming Heterogeneity – the Ptolemy Approach*. Proceedings of the IEEE, v.91, No. 2. Also see http://ptolemy.eecs.berkeley.edu/

Jay W. Forrester. 1961. *Industrial Dynamics*. Portland, Oregon: Productivity Press

Gossink, D., J. Lemmer. 2004. *Recursive Noisy OR – A Rule for Estimating Complex Probabilistic Actions*. IEEE Transactions on Systems, Man, and Cybernetics, Part B.

Pioch, N. J., C. Lofdahl, M. Sao Pedro, B. Krikeles, and L. Morley. 2007. *An Integrated Development Environment for PMESII Model Authoring, Integration, Validation, and Debugging*. SPIE Defense and Security – Modeling and Simulation for Military Operations, Orlando, FL.

Pioch, N. J. 2007. *Commander's Model Integration and Simulation Toolkit*. BAE Systems Advanced Information Technologies, Final Technical Report TR-2067.

Pioch, N. J., J. Melhuish, and M. Cook. 2008. *A Case Study in Integrated PMESII Modeling and Simulation*. 76th Military Operations Research Society Symposium, Briefing for Working Group 1 (Strategic Ops) & Group 8 (Information Ops), New London, CT.

Pioch, N. J., J. Melhuish. 2009. *CyberCAFÉ – Cyber Commander's Analysis of Forecast Effects*. BAE Systems Advanced Information Technologies, Final Technical Report.

Santos E., Santos E. S. 1999. *A framework for building knowledge-bases under uncertainty*. Journal of Experimental and Theoretical Artificial Intelligence, Vol: 11, pp. 265-286.