12th International Conference on Information Fusion
Seattle, WA, USA, July 6-9, 2009

Model Based Integration and Experimentation of Information Fusion and C2 Systems

Sandeep Neema, Ted Bapty, Xenofon Koutsoukos, Himanshu Neema, Janos Sztipanovits, Gabor Karsai
Institute for Software Integrated Systems
Vanderbilt University
Nashville, TN, U.S.A
{sandeep,bapty,koutsoxd,himanshu,sztipaj,karsai}@isis.vanderbilt.edu

Abstract – *Modern Network Centric Operations drive the complexity of Information Fusion and Command and Control (C2) Systems. Driving this complexity further is the interplay dynamics of the human element, information systems, and communication networks. The lack of low-cost realistic experimental context limits the testing, evaluation, and further development of fusion systems to small-scale localized experiments. A Model-based Integration and Experimentation Framework is proposed. This framework is built on C2 Wind Tunnel – a robust multi-model simulation framework for integrating simulations to drive fusion experiments. The second component of the framework is a model-based system of systems integration tool-suite that allows modeling, synthesis, and deployment of networked system of systems. This component enables researchers to embed their research algorithm into the networked C2 systems. The C2 Wind Tunnel has been used to execute several simulation-driven C2 Experiments.*

Keywords: Model Integrated Computing, Service Oriented Architectures, Information Fusion, Integrated Simulation Environment

1 Introduction

Information Fusion is a challenging, mathematically complex, inter-disciplinary research problem. Significant advances have been made by domain researchers in signal processing, artificial intelligence, and data mining to develop methods, and algorithms, for multi-sensor data fusion. However, significant challenges remain, driven by the emerging trends towards Network Centric Operations. The sensor coverage has been increasing, both in military, and in scientific and commercial applications due to reductions in cost and the proliferation of observation platforms. There is an explosion in the scale, and diversity of sensors brought together by networks – wired and wireless. Arguably, increased sensor coverage can improve the situational awareness, however, exacerbates the complexity and workload of fusion algorithms and the network. The variability in terms of availability and bandwidth of networks introduces more uncertainty, driving down the confidence in fusion results. The most significant challenge to information fusion, however, comes from the significant interplay with the human element for situation understanding and assessment, both as consumers of the results of information fusion, and provider of input into the fusion process, either as direct observations, decision guidance, or as data generated by humans.

Traditionally researchers have studied and developed algorithms in isolation. Testing and evaluation of their algorithms is done with pseudo random data and localized experiments. This is driven by the primary need of research community to develop proof of concept experiments, but also the limited availability and cost and effort involved in setting up realistic, context-appropriate experimental scenarios. In a realistic deployment, the fusion algorithm does not operate in isolation, but has to interact with the software, middleware, platforms, networks, and humans. The absence of context-driven experiment, therefore, raises the potential for algorithm-application mismatch, and can hamper the development and refinement of the research. The research algorithm may produce, high quality fusion results, but when not integrated and marshaled appropriately in the deployment context, may not achieve full anticipated operational effectiveness.

Motivated by this lack of a low-cost, affordable, easy to deploy experimentation testbed, we are developing a framework that will enable Information Fusion and C2 researchers to integrate their algorithms, evaluate, and validate in realistic scenarios, supplying target environment reality in terms of simulated environments, human organizational interplay, and anticipated network properties. There are two core elements of our framework: 1) *C2 Wind Tunnel* (*C2WT*) *– a multi-modal simulation integration framework*, and 2) *Model-based Networked System of System Integration tools*. The key characteristic of our approach is the pervasive use of models, and model-based tools, which render the experimentation framework low-cost, and facilitates rapid creation and deployment of experimental scenarios, with minimal manual effort.

Figure 1 shows a notional experiment deployment on our experimental testbed. The Simulated/Emulated C4ISR System represents the Fusion and C2 System, where the researchers integrate their algorithms, while the simulations surrounding the Simulated C4ISR System provide the context.

The rest of the paper is organized as follows: section 2 presents the multi-model simulation environment (C2 Wind Tunnel), section 3 elaborates upon the Model Integration Framework for Networked System of Systems, section 4 presents the approach to scenario driven experimentation,

section 5 provides a case study, and finally section 6 provides a summary and concluding remarks.

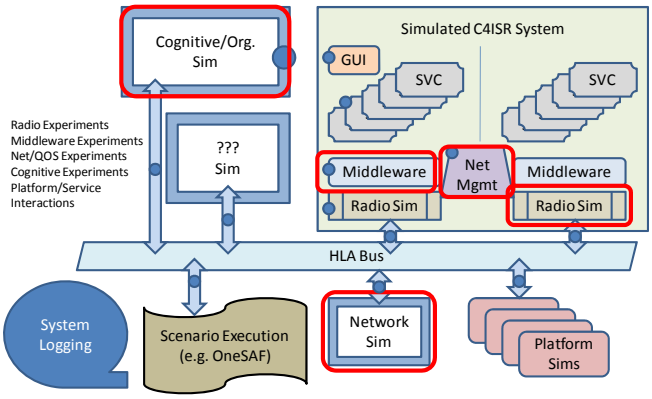

Figure 1: A fusion experiment.

2 Multi-model Simulation Environment

The continuous integration and experimentation of Information Fusion and C2 Systems, requires combining heterogeneous simulation tools and real components, each of which potentially could have its own modeling formalism and an underlying model of computation. Achieving a seamless integration between requires addressing the semantic gap between the models of computation of the different simulators. Furthermore, in an integrated C2 experiment, simulations can have varying degrees of fidelity and operate at different timescales. A 'virtual prototype' of a system can actually be a suite of prototypes, each element representing a system (its hardware, software, users, etc.) and its environment with a different fidelity. The rapid, yet correct configuration of simulations is an essential capability. We have developed C2 Wind Tunnel (C2WT), a model-based simulation integration framework that addresses the challenges listed above.

C2WT addresses the semantic gap problem using discrete event system model (DEVS) as its model of computation. DEVS is considered to be the most fundamental model of computation that subsumes others [15]. DEVS provides a computational model and semantic basis for dynamic system simulation that serves as the foundation for all other modeling formalisms. This observation is validated by the High-Level Architecture (HLA) [16], developed by the DoD: A framework for integrating heterogeneous simulations. In HLA, the framework that regulates and facilitates the interactions among simulation engines is based on concepts similar to discrete-event system models.

C2WT addresses the coincidental complexities inherent in integrating simulators, by pervasive use of models, and providing a model integration layer that enables rapid integration and configuration of simulators in a C2 experiment. The rest of this section elaborates upon the C2 Wind Tunnel.

2.1 C2 Wind Tunnel

The C2WT is an integrated, multi-model simulation environment for the experimental evaluation of congruence between organizational and technical architectures in C2 systems. The C2WT framework uses the discrete event model of computation as the common semantic framework for the precise integration of an extensible range of simulation engines. These simulators are integrated with the Run-Time Infrastructure (RTI) of the HLA platform. Each simulation model, when incorporated into the overall simulation environment of C2WT, requires integration on two levels: the API level and the interaction level. API level integration provides basic services such as message passing, and shared object management, whereas interaction level integration addresses the issues of synchronization and coordination. The C2WT offers a solution for multi-model simulation by decomposing the problem into *model integration* and *experiment integration* tasks. Figure 2 shows the architecture of the C2WT.

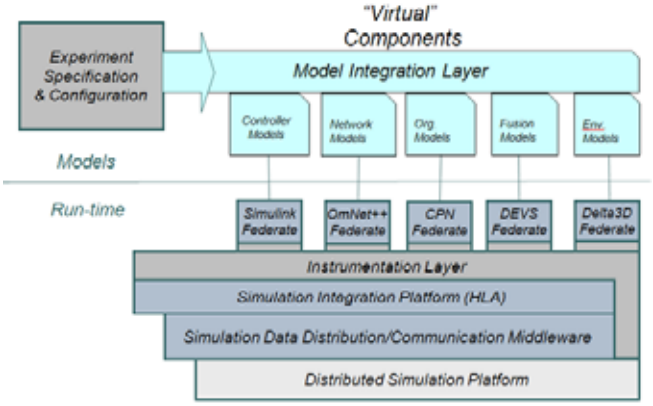

Figure 2: C2WT Simulation Architecture

2.1.1 Model Integration

In C2WT integrated experiments are specified by a suite of domain specific models, including for instance: human dynamics (CPN models) [1][11], networks (OMNET models), physical platforms (Simulink/Stateflow models), and the physical environment (Delta3D models). While the individual dynamics provided by the different simulation models are essential, they are insufficient for the integrated experiments. Their interactions across the component models need to be formally captured and the simulation of the components needs to be coordinated. This is a significant challenge, since the component models are defined using dramatically different domain specific modeling languages (DSMLs). C2WT, therefore, uses metamodeling [2][3][4][5] and the metaprogrammable MIC tool suite [7][10] for developing a Model Integration Layer [9]. A Model Integration Language (MIL) was designed and defined via a metamodel, for formal specification of model integration.

The MIL metamodel consists of a carefully selected collection of modeling concepts that represent the domain-specific simulation tools. Figure 3 shows the primary portion of the metamodel that defines the universe of composition elements of an HLA federation running in the C2 Wind Tunnel framework and the set of communication elements for inter-federate communication. The three main elements in a federation (defined by the model FOMSheet) are Interaction, Object, and Federate representing an HLA-Interaction, HLA-Object, and an HLA-Federate respectively. Federates in an HLA federation communicate among each other using HLA-interactions and HLA-objects – which are in turn managed by the RTI. Interactions and objects, in an analogy with inter-process communication, correspond to the message passing and shared memory styles, respectively. The metamodel fully supports the key attributes of these communication elements such as delivery method, message order (timestamp or receive order), and parameters. The main attribute of a federate, as far as HLA-based synchronization is concerned, is its Look-ahead – the period of time in the future during which the federate guarantees that it will not send an interaction or update an object.

When the MIL domain models are constructed, the designer specifies how the simulations are integrated. The integration models describe both the data representation and data flow elements. The data representation models consist of interaction and object models. Interactions are stateless and can have parameters while objects have states – represented as a set of attributes. Both interactions and objects can have an inheritance hierarchy. These data representation models directly map to the HLA Federation Object Model (FOM). Once the data representation models is created the modeler can define publish-subscribe data flow relations by creating federates and connecting them to interactions or object attributes with directional links. Federates can publish or subscribe interactions or object attributes.

2.1.2 Model-based Experiment Integration

C2WT uses the MIC model interpretation infrastructure (GME, GReAT, UDM, etc.) [6][7] for developing generators that automatically integrate heterogeneous experiments on the HLA platform and deployed on a distributed computing environment. After finalizing the component models, the integration models and setting the parameters, the MIL model interpreters generate all the necessary configuration information. This includes the HLA .fed file that configures the HLA layer; configuration files for the component federates, and where necessary, Java or C++ skeleton code, for run-time use in federates.

2.1.3 Customization Levels of the C2WT

The system has been designed to serve as an extensible infrastructure for conducting computational experiments. C2WT can be customized/extended on three levels: *infrastructure*, *mission,* and *experiment,* as depicted in Table 2-1. Extension of the *infrastructure* means the addition of new simulation (and analysis) tools with their domain specific modeling languages (DSMLs). For example, the integration of MATLAB/Simulink in the C2WT infrastructure is an infrastructure-level customization, as it requires the introduction of the model element in MIL to represent the MATLAB/Simulink engine. *Mission* scenarios are defined with a specification of vignettes that capture the essential aspects of missions. For example, one of the vignettes developed demonstrated in C2WT focused on finding, tracking, and acting on a time critical targets in an urban setting. The scenario is precisely specified using organizational, network, platform and other component models using the appropriate modeling languages and the integration models. Mission scenarios define a framework for configuring and executing *experiments*. Experiment execution can be highly automated after specifying configurations and parameter ranges for component models.

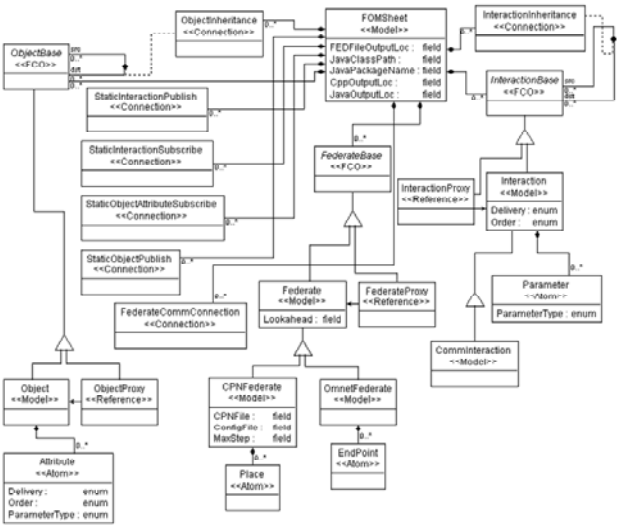

Figure 3: C2 Wind Tunnel MIL metamodel

2.1.4 Time Management in C2WT

Time Management is critical to preserve causality with simulations operating at different timescales. The C2WT builds upon the time management features of the underlying HLA standard, which has provision for both discrete time and discrete event models. The main elements of time management in HLA are: a) a Logical Timeline, b) Time ordered delivery of interactions between simulations, and c) a protocol for advance of Logical Time. In a causality preserving execution (note that HLA supports untimed executions as well), the underlying RTI maintains a logical time, and interaction messages generated by simulations are time stamped with the logical time, and delivered to their destinations in a timed order. The logical time is advanced by a cooperative Time Advance Request

Table 2-1 – Customization Levels in the C2WT

Level	Example	Expertise	Function	Frequency
Infrastructure		Metamodeling. Model transformation.	Adding new model types, integrating new simulation tools.	Incorporating a new discipline or model type in the infrastructure.
Mission		Interaction modeling. Component modeling.	End-to-end modeling of mission scenarios.	Startup activities for new studies.
Experiment		Component modeling. Experiment design. Data analysis.	Configure and parameterize component models for individual experiments and measure selected performance data	Performing studies and evaluations by running experiments

and Grant Protocol. A similar protocol is supported for event driven simulation, in which the event driven simulation requests the Next Event to the RTI. The simulation logical time is advanced either to the earliest available interaction, or to the time stamp of the next event local to the requesting simulation.

The C2WT has been successfully used in demonstrating and conducting simulation based experiments with several military relevant scenarios [18]. The C2WT can also be used as a simulation driven testbed to stimulate, exercise, and evaluate, operational C2/C4ISR systems and software. The challenge inherent in integrating operational software with simulation is the synchronization with real-time, which can be problematic when integrating with slower than real-time simulators. One potential approach that we have considered is *time dilation*, i.e. to slow down the virtual clock as seen by operational software components.

3 Model-based Integration

Modern Information Fusion and C2 Systems for Network Centric Operations (NCO) are Large Scale Networked System of Systems. Metadata is the currency of exchange at all layers and across layers of NCO. In Joint or Coalition Task Force environment, there are multiple Communities of Interest (COI), each one with its own variant (dialect) of the metadata: (a) the common elements – the vocabulary – defined by the DOD-CIO; (b) the common Army metadata; and (c) the unique metadata associated with a particular COI, such as J3C-TIME being developed by CERDEC. Further, there is metadata that can describe organizational structures (MSDL, DyNetML, DAML), information models and flows, communication network topologies etc. Significant challenges need to be addressed to enable precise specification of the metadata, beyond the standardization of data interchange formats.

We argue for a model-driven software development process, where the systems are designed and constructed based on models which are the 'blueprints' that represent the architecture and the salient properties of the system. NCO necessitates a *system of systems*, connected via a multitude of networks. In order to engineer and analyze such ensembles, two major integration problems must be solved: (1) the systems need to be able to communicate and exchange information with each other, and (2) the models of the systems need to be integrated, so we can analyze and understand them. The first problem is addressed via the metadata mentioned above, such that data always has a description that is shared between the systems, but the second problem is a major challenge for large-scale model-driven software development. In short, the problem of *model interoperability* must be solved. The problem is especially acute when models of other domains, like human organizations and decision making processes interact with models of software systems.

Model Interoperability requires significant progress in the explicit representation of the semantics of domain-specific modeling languages (DSML). Composability and customization of integration tool chains (such as modeling, verification and testing tools and virtual prototyping environments) require semantically rigorous model and simulation integration. The following core problems must be solved:

Compositional metamodeling. The precise specification of behavioral semantics for domain-specific modeling languages (DSML) is an ongoing research topic in model-based design. Techniques for metamodel

composition for behavioral semantics [3][12], are required to support construction of new DSML-s by composing existing metamodels.

Compositional construction of model transformations. Model transformations are essential ingredients in any model-integration process: they provide the linkage between various models [13][14] Their correctness is of utmost importance for obvious reasons: incorrect transformations lead to dysfunctional model integration. Techniques are required for building complex, yet validated model transformations from well-defined, reusable, and well-analyzed steps, in a compositional manner.

We are addressing these challenges, by developing a suite of modeling paradigms for the specification and analysis/experimentation of fusion systems. Specifically, these include a system specification, human modeling, and simulation infrastructure modeling. The systems language allows representation of the target system that will contain the fusion system, including the computers, software, and networks in the C2 system. The concepts are briefly described below.

The C2 system must capture multiple views of the target. These views consist of:

- The target network architecture, including the terminals (Receivers/Transmitters), communications paths and topologies, radio waveform properties, and associated bandwidths.
- The target computational nodes, including processors, memory, and resulting computational capability.
- A software view captures the various processes, data interactions, interprocess logical topology, and execution properties of each process. Limited behavioral models describe the rough operations of the process.
- A data view captures the data structures used within the processes and which are transmitted across communications interfaces.
- Deployment models capture where software is placed upon the hardware, defining a specific system instance.

We have developed multiple generation tools that can synthesize significant aspects of the target system. These target systems can be purely simulated, for example in a discrete event system, to get estimates of overall system timing. Alternatively, the software can be targeted at a specific operating system/middleware combination to build a functional prototype of the system. Hybrids of these can also be used, to simulate at low resolution a scaled up scenario, with a few high-fidelity segments that execute the full behavior of a part of the system.

Given these tools, researchers can rapidly integrate many different experiments at a low cost (in terms of manpower). This can provide the mechanisms for a researcher to scale up an experiment, try different stress-test cases, network topologies, etc. Further, researchers in the Social Cognitive, Information, and Communication Networks layers develop their own modeling languages (DSMLs) and models using distinct syntax and semantics. The current approach uses timed colored petri nets to capture human and organizational interactions. These can be tied into system computational events to capture the influence of the user on the system and the influence of the system on the user.

Metamodeling enables us to precisely "model the individual DSMLs", compositional metamodeling allows formally integrating the DSMLs into heterogeneous system modeling languages and compositional construction of model transformation enable us to translate the heterogeneous models into languages defining common semantic domains such as discrete event dynamics or hybrid automata.

Figure 4 shows C4ISR System service layers, each of represents a placeholder for researchers to integrate their algorithms. The model generation approach synthesizes integration interface of the researcher's algorithm with the appropriate middleware, enabling the researcher to focus on the core algorithmic issues.

4 Scenario-Based Experimentation

Scenario-based experimentation drives the system under test with a series of operationally relevant scenarios or vignettes [8]. Specific parts of the problem are cast into the capabilities of the research components. Information exchanges and interfaces are defined, along with a general process or interaction diagram. Challenges abound in performing scenario based experiments, which are compounded by the scale of the experiments:

- Researchers must "speak the same language" and understand the requirements and capabilities of other components.
- Semantically sound integration and coordination of simulations and emulations, operating at different time scales and on different assumed data syntax & semantics need adapters to provide the glue between different software entities. (We cannot force a common data interchange standard)
- Configurations of simulations for a mission scenario must be developed to provide correct stimulus
- Deployment of simulations over computing clusters and configuration of the coordination mechanisms, and a runtime system is needed to configure the system of systems,
- Configuration of the data collection for analysis and computing metrics is needed to measure system performance and give feedback.

In current practices, specification and performance of such experiments takes months, requiring large teams of developers and integrators. This large cost limits the scope of the scenario, and the exploration of variation points and what-ifs. C2WT assists in overcoming these challenges. Pervasive use of models offers a common set of semantics for describing component interaction (a common

language) and high degree of automation in conducting such experiments. This reduces the experimentation time from months to day, and weeks to hours, enabling a much larger set of experiments to be executed.

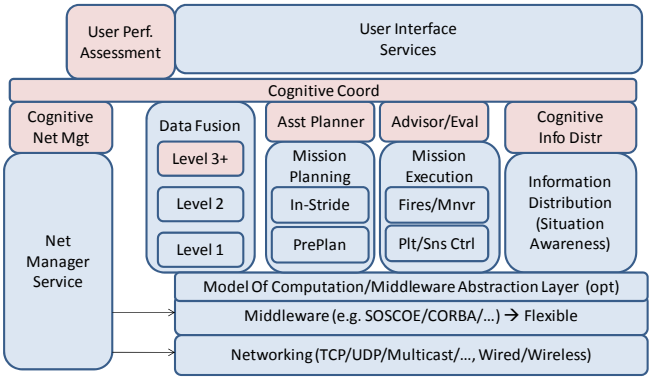

Figure 4: C4ISR System Service Layers

4.1 Experiment Design Specifications

The following activities must be undertaken to define and execute scenario based experiments.

Identify models, simulations, and emulations: for simulating (emulating) different aspects of the scenario and providing valid system-under-test stimulation, such as:

- Virtual Environment Simulation: Delta-3D, Ogre-3D, OneSAF, for, force-on-force interactions, terrain, and maneuvers, with the Military Scenario Description Language (MSDL), which captures the Unit Task Organization and Locations.
- Platform Simulations: Mathwork's Simulink entities, military vehicle sims, low-resolution platform sims, to orchestrate the movement of platforms either scripted, or interactively guided.
- C4ISR Software Simulation/Emulation: Components to implement a C4ISR system can come from many different sources, and should be available for the target system hosting a fusion algorithm.
- Network Simulation: NS-2, Omnet++, Opnet, and other network simulations tools to accurately model communication network jitter and packet losses, and to assess the impact on information and social cognitive networks.
- Cognitive / Org Analysis and Simulation: CMU's SORASCS, can be used for human behavioral modeling and socio-cognitive networks, with additional services from the information, socio-cognitive and communications level. Integration with SORASCS addresses an important problem: full-spectrum reasoning requires that experimental testbeds operate at two levels – short duration real time for command and control analysis, and long duration for reach-back analysis where the focus is on fusion, gathering, cleaning etc.
- Additional Simulation tools, developed or provided by the research community, and can be added and incorporated in a library of configurable simulation and emulation tools.

Specify interactions between simulators: The Model Integration Language (MIL) developed in C2WT is augmented and used for specification of interactions between simulators.

Configure the simulators: Simulator specific data and configuration files are initially developed per experiment. As our framework evolves, data will be available from a model/data repository. We use existing ontologies (e.g. MSDL for Task Org and Layout) and data standards (DTED, 3DS, MDL) wherever possible for such specifications. Furthermore, an extensible library of models can be developed, that can be rapidly used and adapted to configure the simulation tools for a specific mission scenario.

Specify the deployment of simulators: The Model-Based Experiment Integration Language, developed in C2WT is also for specification of experiment execution, specifying the computing resources, and the deployment of simulation tool on computing resources. Researchers can leverage clusters of Emulab [17], and VMWare based virtualized experimentation infrastructure, to perform large scale deployments (100-s of nodes).

4.2 Experiment Monitoring

Our framework provides facilities for monitoring the status of ongoing experiments. This includes diagnostics information about the health of various simulators, health of the compute nodes, health of the communication links, processes, etc. The framework also provides facilities for recording of selected data streams from the simulations, automated by C2WT tools. The recorded data streams are playback ready, enabling a variety of experimentation scenarios: 1) Repeat of Full experiment with different configuration for conducting "what-if" experiments, 2) Partial experiments – only a subset of simulators are run, and the rest are substituted by playback tools that simply playback data that was recorded earlier, which enhances the scalability of the experimentation. The framework can be augmented with additional provide facilities for the management and configuration control of experiments conducted in the Integration Centers, including tagging and archiving of experiments, experiment configurations, models, data, documents, and experiment results.

5 Case Study

We developed a realistic command and control scenario involving tactical and operational decision making in support of C2 operations in the presence of an active adversary in a contested cyber environment. The scenario not only exemplified the multi-model simulation, but also demonstrated resilient C2 under a cyber attack. We refer to the human command and control intelligence

organizations as the "Blue" team and the intelligent, adaptive adversary organization as the "Red" team. This is shown in Figure 5 below.

In this scenario, Red operations involves setting off Improvised Explosive Device (IED) vehicles in an urban city area. They have a safe house at their disposal where they manufacture bombs – called bombfactory. They receive large shipments of bomb materials at the factory periodically. A RedLeader agent controls actions of Red actors. RedLeader informs the bombfactory when the shipment is about to be delivered.

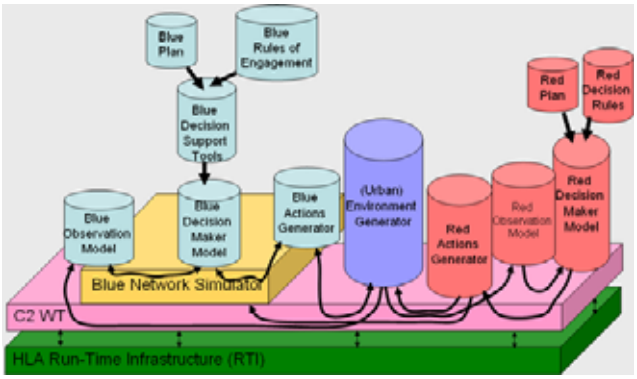

Figure 5: Blue and Red Actors in a C2 scenario

On the other hand, Blue team has two Unmanned Aerial Vehicles (UAVs) at their disposal to perform area inspections. They also have prior knowledge of some aspects of how Red operates. Also, one of the Blue actor is a Signal Intelligence (SIGINT) division that can listen on cell phone communications in a given geographical area in order to localize certain Red actors.

The scenario begins by Blue team receiving a tip-off from a RedInsider agent about the description of a suspected bomb materials delivery truck going to the bombfactory. Blue team first commands UAV-1 to FFT (find, fix, and track) suspected bomb materials delivery truck. When the truck enters the urban area where they suspect possible location of the bomb factory, Blue team initiates SIGINT to monitor cell phone calls in that area. At this time, RedLeader informs the bombfactory about the arrival of the bomb materials delivery truck. This leads to the knowledge of bombfactory location by the Blue team. In response, Blue team targets UAV-2 to monitor bombfactory area to see if suspected IED vehicle (typically a small pick-up truck) leaves the bombfactory. When such an IED vehicle departs, the Blue team concludes possible IED attack and commands UAV-2 to locate IED location. Once the truck stops at IED location, SIGINT reports about another Red cell phone call made to the Red Lookout agent for initiating the explosion. The scenario ends with successful identification of bombfactory, IED location, and Lookout location.

To faithfully simulation this command and control scenario, we needed to build and integrate several different models developed using different simulation tools. We used C2 Wind Tunnel for integrating various simulation models as it comes with *out-of-the-box* support for integration of a wide array of simulation tools viz. MATLAB/Simulink, CPNTools, OMNeT++, C++/Java based tools, DevsJava, Ogre3D, Delta3D, Google Earth 3D, etc. For example, we used full-scale UAV dynamics models developed for four-rotor helicopters by Berkeley using MATLAB/Simulink models. Operations of both the Blue and Red teams were modeled using Colored Petri Net (CPN) models built using CPNTools. The UAV operator models were again developed in Simulink. For network simulation we used OMNeT++ and INET Framework models. Also, the 3D visualization for UAV fields of view were done using Google Earth 3D imagery. For demonstration we designed the scenario in an actual city area with trucks moving along highways and UAV showing the real 3D buildings while tracking the trucks. Figure 6 below shows screenshots from a demonstration of this scenario.

We also developed a number of scenario excursions (what-iffs) beyond the main success scenario. One of the excursions involved a cyber attack on the downlink used by UAVs to trasmit images from their current field of view to the ground station UAV operators. Under these circumstances, the UAV operator was obviously not able to control the UAVs thus causing a mission failure. In yet another similar excursion, the SIGINT detects the cyber attack and informs Blue's Cyber Cell Division which – after certain time-period performs Anti-Jamming (AJ) to restore UAV downlinks. In this case, the UAV operators are again able to guide the UAVs per the scenario. However, we show that owing to the delay caused by the cyber attack, despite successful scenario execution, the overall mission performance does get impacted.

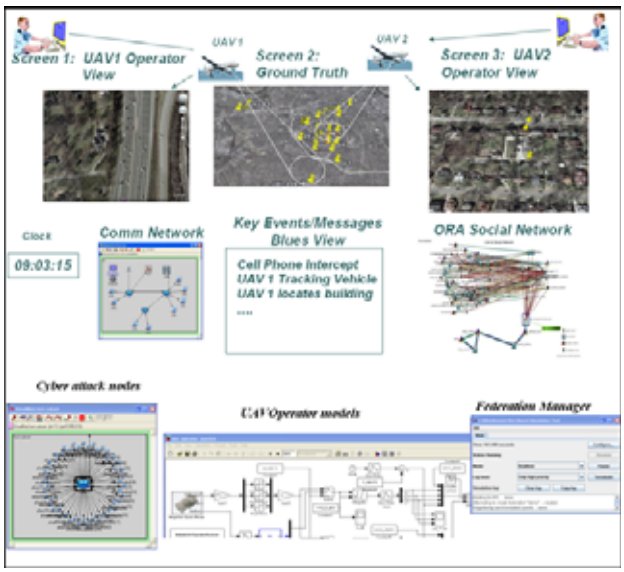

Figure 6: Cyber scenario using C2 Wind Tunnel

6 Conclusions

In the paper, we briefly, described a Model-based Integration and Experimentation framework that enables Information Fusion and C2 Researchers to integrate their algorithms and assess, evaluate, and validate in realistic scenarios. Our framework is built upon C2 Wind Tunnel (C2WT) – a robust multi-modal simulation integration framework – that allows integration of simulation and live systems, and a Model-based Networked System of System Integration framework that enables researchers to model their research algorithm and synthesize and integrate with a distribution middleware. The different components of the framework described above are in different states of maturity. The C2WT multi-model simulation framework is very robust and has been used in orchestrating a wide variety of simulation based experiments. The Model-based Integration framework for Networked System of Systems has been developed for a specific proprietary middleware, and needs some effort in generalizing and adopting a different open-source middleware. However, the core model-based technologies (Meta-programmable GME, MIC) used in both these components are highly sound and robust and are used world-wide.

References

[1] A. H. Levis, "Executable Models of Decision Making Organizations," in Organizational Simulation, William B. Rouse and Ken Boff, Eds., Wiley, NY, 2005.

[2] Emerson M., and Sztipanovits J., "Techniques for Metamodel Composition," OOPSLA – 6th Workshop on Domain Specific Modeling, Portland, Oregon, October 22 2006, pp. 123-139.

[3] Jackson, E., Sztipanovits, J., "Towards A Formal Foundation for Domain Specific Modeling Languages," Proceedings of the Sixth ACM International Conference on Embedded Software (EMSOFT'05), Seoul, Korea October 22-25 2006, pp. 53-63.

[4] Jackson, E., Sztipanovits, J., "Formalizing the Structural Semantics of Domain-Specific Modeling Languages," Journal of Software and System Modeling (SoSYM), to appear in 2008.

[5] Jackson, E., Sztipanovits, J., "Constructive Techniques for Meta- and Model-Level Reasoning," Proceedings of Models 2007, LNCS 4735, 405-419.

[6] Karsai, G., Agarwal, A., Shi, F., Sprinkle, J.: On the Use of Graph Transformation in the Formal Specification of Model Interpreters, Journal of Universal Computer Science, Volume 9, Issue 11, 2003.

[7] Karsai, G., Aditya Agrawal, Zsolt Kalmar, Sandeep Neema, Feng Shi, Attila Vizhanyo: "The Design of a Language for Model Transformations", Journal on Software and Sys-tem Modeling pp. 261-288, Volume 5, Number 3 / September, 2006.

[8] V. Y. Jin and A. H. Levis, "Command and Control Experiment Design Using Dimensional Analysis," Proc. 1988 Symposium on C2 Research, Monterey CA, June 1988.

[9] H. Neema, G. Hemingway, J, Green, B. W. Williams, J. Sztipanovits, G. Karsai: "Rapid Synthesis HLA-Based Heterogeneous Simulation: A Model-Based Integration Approach", ISIS Technical Report,May, 2008.

[10] Karsai, G., Ledeczi, A., Neema, S., Sztipanovits, J.: The Model-Integrated Computing Toolsuite: Metaprogrammable Tools for Embedded Control System Design, IEEE Joint Conference CCA, ISIC and CACSD, Munich, Germany, 2006.

[11] A. H. Levis, S. K. Kansal, A. E. Olmez, A. M. AbuSharekh: Computational models of Multi-national Organizations, in *Social Computing, Behavioral Modeling and Prediction*, H. Liu, J. L. Salerno and M. J. Young, Eds., Springer, 2008.

[12] K. Chen, J. Sztipanovits, S. Neema: "Compositional Specification of Behavioral Semantics," Proceedings of DATE 2007, pp. 906-912, Nice, France, April 20-25, 2007. (Best Paper Award, Selected for The Most Influential Papers of 10 Years of DATE)

[13] G. Karsai, "Design Tool Integration: An Exercise in Semantic Interoperability," Proceedings of the IEEE Engineering of Computer Based Systems, Edinburgh, UK, March, 2000.

[14] Karsai G., Lang A., Neema S.: Tool Integration Patterns, Workshop on Tool Integration in System Development, ESEC/FSE, pp 33-38., Helsinki, Finland, September, 2003.

[15] H. Vangheluwe, "DEVS as a common denominator for multi-formalism hybrid systems modeling", In Andras Varga, editor, IEEE International Symposium on Computer-Aided Control System Design. Anchorage, Alaska, September 2000, 129-134.

[16] J.S. Dahmann, R.M. Fujimoto, R.M. Weatherly, "The Department Of Defense High Level Architecture," ,pp.142-149, 1997 Winter Simulation Conference (WSC'97), 1997.

[17] E. Eide, L. Stoller, and J. Lepreau. "An Experimentation Workbench for Replayable Networking Research," In Proceedings of the Fourth USENIX Symposium on Networked Systems Design and Implementation (NSDI 2007), pages 215-228, Cambridge, MA, April 2007.

[18] C2WT Wiki, C2 Wind Tunnel Project Website, https://wiki.isis.vanderbilt.edu/c2w/index.php/Main_Page, 2008.