New capabilities and features of the C2 Wind Tunnel

Author: Himanshu Neema
Date: 04/02/2009

Below, I list the new capabilities and features of the C2 Wind Tunnel framework. With these new features, C2 Wind Tunnel has truly evolved into a platform for performing and analyzing everyday command and control experiments:

1. Photo-realistic 3D visualization of urban environments using Google Earth

We have integrated Google Earth's 3D visualization for accommodating urban environments in the simulation. This supports using photo-realistic 3D imagery. We simulate the live streaming of images from UAV sensors on to the operator consoles.

For the integration of Google Earth based 3D visualization, we developed newer federates that can perform Physics calculations and play the role of operator consoles. The physics federate keeps track of all aerial and ground vehicles, their orientation, direction of movements, speed, etc. In addition, it also performs calculations for determining the closeness of two moving objects. This information is transferred to operator console federates over the network -- which is again a newly developed fairly detailed and involved federate module as explained below.

We also developed a “Ground Truth” view that shows a bird's eye view of all the entities in the simulation. For example, it shows a top-level 3D view of chosen urban city environment, with UAVs flying in the air and trucks moving on highways and local streets.

2. Network simulation

We have also developed detailed network simulation in support of the C2 experiments. This involves simulating the streaming of data with delays that occur while data is transmitted over the network. We used OMNeT++ network simulator to encode the network behavior while faithfully simulating the entire network stack. We also simulate network attacks, cell-phone communication, and cell call interception by SIGINT as described later in the document.

3. Models for human-organization decision-making process

We have completed human organization decision models using CPN tools. These models are used for the intent, decision-making process, and actions of both Blue and Red teams in our scenarios.
4. Cyber affects with Distributed Denial of Service (DDoS) attacks

We have also modeled a Distributed Denial of Service (DDoS) type of attack that can completely block the downlink between UAV sensors and the operator console. The result of this attack is that the images on the operator console freezes for the duration of the attack due to which the UAV operators no longer remain able to control the UAVs.

5. Cell phone communication

We have also modeled cell phone communication among Red and Blue team players. These cell phone calls are triggered dynamically based on events in the simulation and contain directive information that further changes the execution of the scenarios.

6. Models for Signal Intelligence (SIGINT) division and build-up of social networks

Using the above DDoS type cyber affects, we also model the Signal Intelligence (SIGINT) division as well as how it reverses the effect of jamming. We show that under certain circumstances, when the Blue team can successfully perform anti-jamming (within a reasonable amount of time), missions can still be accomplished avoid failures though with decreased mission performance. We also show the catastrophic consequences of not being able to negate the jamming successfully.

Perhaps the main role of SIGINT is to intercept the cell phone communication between team players. When tasked by the ISRD division, SIGINT may start monitoring certain geographical area for cell phone calls between suspected Red team players. When such a call is intercepted (using the network simulator), the Blue team can update its information and react accordingly.

We also used a tool called ORA (Organizational Risk Analyzer) to display how the intercepted cell phone calls can be used to build social networks that exist among the Red team players and use this process of ever increasing knowledge of social networks to fill holes in the InfoBase (i.e., connect the dots).

7. Federate playback using mock-up engine

We have added a technique to write a mock-up version of a federate (with a corresponding mock-up execution engine) that can be used to replace real federate models. This is a great tool for being able to test newer models and techniques in our federates, as the other part in the federation can easily be played back using this technique without requiring a full set of hardware.
8. Human-in-the-loop functionality

For demonstrating human-in-the-loop, we developed UAV operator models using which real humans participate with the live simulation. In this model, we interfaced 3Dconnexion joysticks so that humans could actually operate the UAVs as they fly in an urban environment. The human operators have full eight degrees of freedom when guiding the UAVs, viz. left, right, front, back, up, down, yaw, and pitch controls.

While the human UAV operator drives the UAVs, we show the UAV's Field Of View (FOV). In the UAV's field of view, we visualize the 3D display of real world using Google Earth.

9. Support for Customization, Deployment, and Experimentation

We have developed a new version of the GME-based model integration environment. In addition to the models of integration of variety of federates we have further extended the modeling paradigm to support automatic deployment of these federates on the modeled hardware (which maps to real hardware to run the simulations on).

Right from the model the designer can design experiments where they can choose which federate to run and with what parameters. They can define the hardware available for simulation experiments with their platforms (Windows or Linux), their IP-addresses, etc. In addition, deployment models can be created where we can choose which federates to run on which piece of hardware. This allows experimenter with ease of running experiments with the available set of hardware without effort. They simply design the experiments, hardware available, and a deployment model and the GME tools automatically generate the scripts to perform the corresponding execution. In particular, it uses a 'ssh' based login mechanism to copy over the needed files to remotely run processes. Now, after the deployment scripts are generated, an experiment can be run with the push of a button as well as stopped on all machines with the push of a button.

10. Non-invasive integration of custom (user-defined) simulation environments

We have opened the C2 Windtunnel framework to support non-invasive integration of custom (user-defined) simulation environments. Essentially, this means extending the Windtunnel platform itself with custom simulation environments that are either not supported in the Windtunnel (e.g., Java, C++, CPN Tools, Matlab/Simulink, Devsjava, etc), or are of proprietary nature, or simply that cannot be shared in public due to confidentiality concerns. The support for this integration is open in such a way that it does not require you to re-compile the released Windtunnel software. This is accomplished by using Java reflection APIs.
11. Autonomous UAV operation/control

We also added functionality for autonomous UAV controllers. This uses sophisticated controller models (in Simulink) for being able to find, fix, and track the trucks automatically without the use of a human operator. This has been demonstrated in one of our demo scenarios. Also, we have a fuel gauge for UAVs, which can allow us to compare manual versus autonomous UAV operation.

12. Data collection and Logging support

We also recently augmented the C2 Windtunnel modeling framework for major configurability and flexibility enhancements. It is become much easier to use the tools to design and deploy experiments. Several new features have been added to identify problems with the model and to fill in the parameters that can be derived automatically from the current models. The framework now supports a variety of logging and data collection points. This logs are saved to a MySQL database for each experiment run by the user. User can choose to log:

- Whenever a federate sends an interaction,
- Whenever an interaction is sent by any of the federates,
- Whenever a federate receives an interaction,
- Whenever a federate updates an attribute of an object,
- Whenever an object is updated by any of the federates, and
- Whenever a federate receives an update to an object

Apart from above examples of data collection, we also collect data from the re-usable code that is used in all federations. This can be extended at will to log data to any level of details we want. As an example, a FINE level will log almost everything that happens during a simulation, while an IMPORTANT level will log only the important events in an experiment. Again, these logs are also saved to a database.

13. Importing existing Federation (.fed) files for legacy FOM/SOM

We have also extended the C2 Windtunnel modeling framework to be able to import an existing federation file. This is especially helpful for people who have a variety of legacy FOM/SOM files at their disposal and need to migrate them to C2 Wind tunnel. The automatic import tool lets them import a federation file and automatically draw the corresponding models in GME (e.g., federates, interactions, objects, publish and subscribe relationships, etc.).

User can then extend this models by defining a deployment model and leverage all the capabilities of C2 Wind tunnel from there on including automatic glue code generation for integrating federate models with HLA, deploying federates automatically on chosen remote machines, running experiments, collecting data, analyzing data, and so on and so forth.
14. String-Template based code generators

We have completely revamped the code generator mechanism in the C2 Wind tunnel framework. We now use sophisticated string template library for generating code. These not only run very efficiently and are much easier to use, but at the same time these are much easier to maintain while the framework evolves over time.

15. Integration of Spreading Activation Networks (SA-NET) Partial Order Planner (SA-POP) tool

We have integrated the Spreading Activation Networks (SA-NET) Partial Order Planner (SA-POP) as a federate in the C2 Wind Tunnel framework. This tool provides a dynamic decision making tool to support human decision makers as they monitor the progress of evolving missions, and make decisions on what goals to continue to pursue, whether to allocate or reallocate resources and assets, and whether operations need to be scheduled or re-scheduled as the mission advances.

SA-POP operates at a more abstract symbolic level, continually getting input from the scenario as it evolves, assesses the feasibility of current plans and operations in light of changing constraints within the scenario, or as new situations occur. It makes suggestions to the human operator on the feasibility and likelihood of success of various partial plans that are currently being executed, and suggestions on how alternate plans may improve the chances for success. The final decisions on changes are left to the human operators and supervisors.

16. More support for Cyber affects use-cases

C2 Wind Tunnel has now developed into a platform for running a variety of experiments. We intend to develop several use-cases that provide a number of examples of kinds of things we can do with the Wind Tunnel. Toward that end, we have added several capabilities in our models. Being able to jam a network with DDoS type of attack is one of them. Several newer features have now been added to enable demonstration of these use-cases. For example, while the simulation is running:

- UAVs can be switched from being manually operated to being autonomously driven,
- UAVs can be tasked to track a different target,
- UAVs can even be tasked to go after another UAV (a feature that can support a use-case of UAV compromise by the Red team),
- UAVs can perform hand-offs among themselves to re-assign targets, and
- A new attack module can be initiated that prevents (or make arbitrarily difficult) UAV operators from being able to control the UAVs.

We envision that these use-cases will make a strong case for performing command and control experiments using the C2 Wind tunnel infrastructure.
**Current focus:**

The above listed features and capabilities provide a strong platform for performing command and control experiments. We are currently working on further extending the capabilities of the C2 Wind Tunnel Framework toward much better experimentation, data-collection, and analysis support.

We are also working on developing tools to post-process the data collected into variety of ways. Not only, we plan to use WebTAS for data visualization, we also plan to develop custom scripts/procedures to generate a variety of insightful reports from the collected data.

We are also working on parameterized scenario definition for batch execution of a variety of experiments and collect data for various parameters set dynamically for the experiments.