Lowering the Barrier to Advanced Simulation Analysis

USING PYTHON INTEGRATION TO AUTOMATE JCATS ANALYTICAL RUNS

Mr. Stefano Izzo, Mr. Casey Wilson,
COL George Cushman, MAJ Weston Benson

The Problem

Computer based simulations can be limited in supporting military operational analysis or COA evaluation…

- Database
- Time Requirements
- Micromanagement

A Solution: JCATS Python Gateway

- **The JCATS Python Gateway is an interface between JCATS and user created Python code. It provides powerful new options to model complex, custom behaviors & scenarios in ways never imagined by the original JCATS architects.**

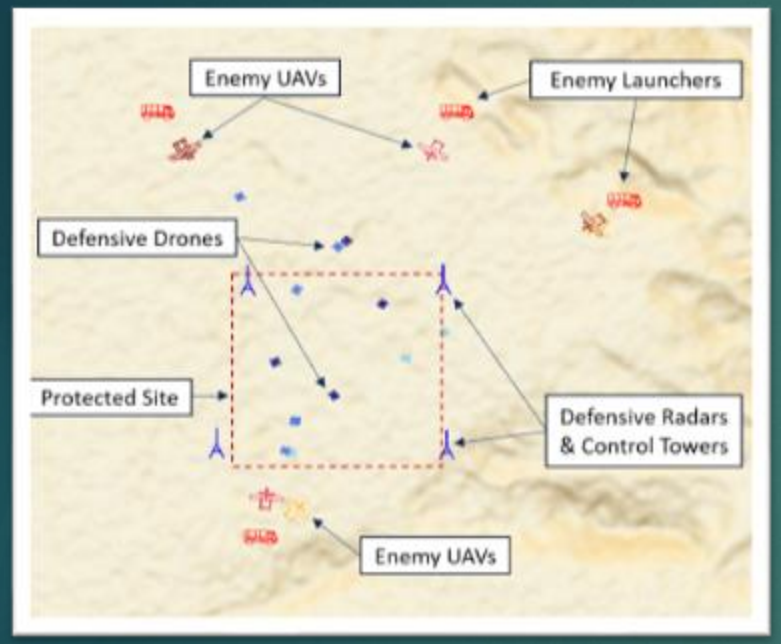
- Users apply custom python code to enable complex entity behaviors and monitor states
- Users can leverage the power of python, external libraries, and can even use python to connect to external software
- The Python Gateway can be used in conjunction with JCATS BatchExec to run analysis scenarios in batch mode

- JCATS provides code hooks for event triggers

Trigger Types:

Time, Movement, Mounting, Acquisitions, Enemies, Sounds, Chemicals, Engagements, Detonations, and more

Use Cases: Experiment and Analyze



HPM Experiment

Leveraging advanced behaviors to test theoretical C-UAS Systems

Helped inform further modeling and simulation focus and future system development

Drone Curtain for Point Defense

Integrating prototype drone swarm control software for further modeling and simulation

Greater flexibility with area of operations and threat systems

Use Cases: Military Operations

REARWARD PASSAGE OF LINES

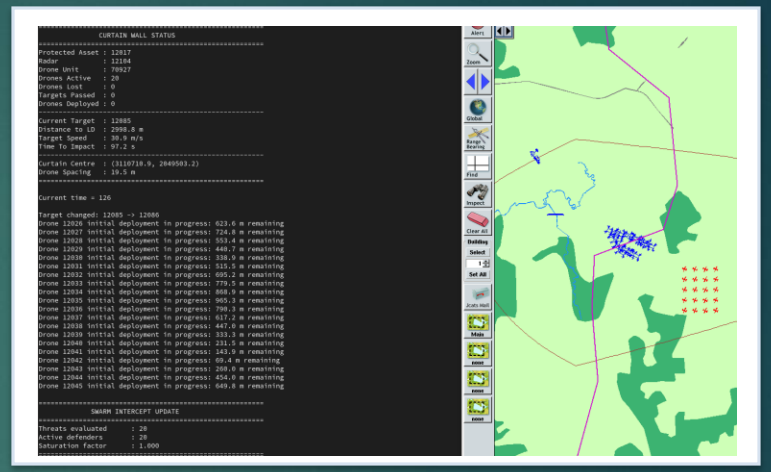
Monitor unit health and trigger withdrawal / passage behavior based on battlefield conditions.

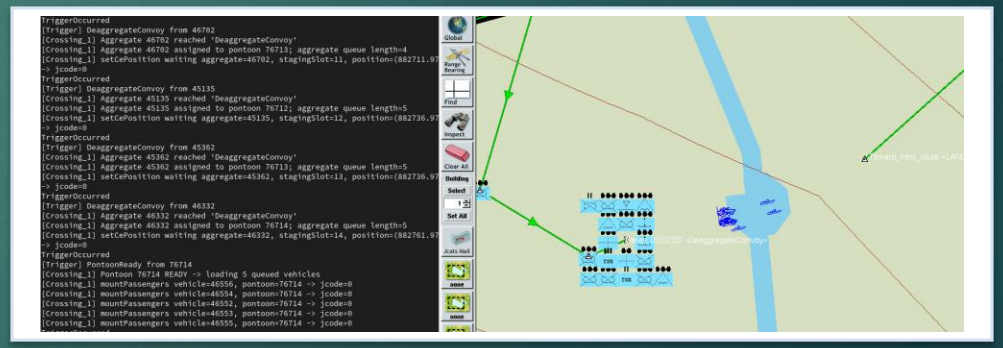
DELIBERATE DEFENSE - CURTAIN CALL

Dynamically calculate and re-position a drone screen against incoming threats; model interception and defender attrition.

WET GAP CROSSING

Distribute formations across available pontoons, queue traffic, reroute when crossings become unavailable, then re-aggregate.





JCATS environment + Gateway control + Python logic + AI-assisted development

What AI + Python Changes

LESS MICROMANAGEMENT

Operators define intent, conditions and decision rules instead of continuously controlling individual entities.

FASTER DEVELOPMENT

AI-assisted coding accelerates translation of an operational idea into Python logic, debugging, testing and iterative refinement.

NEW BEHAVIORS

Gateway control lets entities react dynamically to simulation state rather than only execute predetermined movement plans.

Operational Idea → Natural-Language Discussion → Python Logic → Gateway Commands → JCATS Behavior → Test & Refine

AI lowers the technical barrier between an analyst's operational idea and the code needed to express it in JCATS.

Way Ahead

- The Sim Gateway is well-aligned to integrate with many common Python-based AI libraries in the future.
- Future work includes leveraging this capability for AI integration to model OPFOR based on doctrine and relevant Tactics, Techniques, and Procedures.

QUESTIONS?