U.S. ARMY COMBAT CAPABILITIES DEVELOPMENT COMMAND DEVCOM ANALYSIS CENTER

The Tactical ISR Performance Suite (TIPS) within FRACTALS, A Framework for Capability-based Tactical Analysis Libraries and Simulations

Eric Harclerode

M&S Technical Specialist

C4ISRD, ISR Branch

TACTICAL ISR PERFORMANCE SUITE (TIPS)

TIPS: a modular code library that encapsulates DAC's sensor performance and intelligence process *capabilities* (algorithms and methodologies).

- **Features:**
  - Sensor decomposition into Field of View, performance models, and measurement model;
  - Information processing models (e.g., data fusion, direction finding, other PED processes)
  - Custom-resolution behaviors and processes (e.g., search and acquisition behaviors, dynamic tasking)
  - Varying fidelity performance model options for EO/IR, Radar, SIGINT, and other sensors modalities.
- **Implementations:**
  - *FRACTALS* (Tactical Simulation Framework / Visualization Tool)
    - Enables rapid sensor system prototyping/modeling through a robust user interface
    - Provides visualization tools for sensor search patterns and coverage
  - Item-level Data or Sensitivity Analysis Tool (Sensor Performance)
  - External Applications – Multi-domain Sensing Architecture (M&S as a Service)

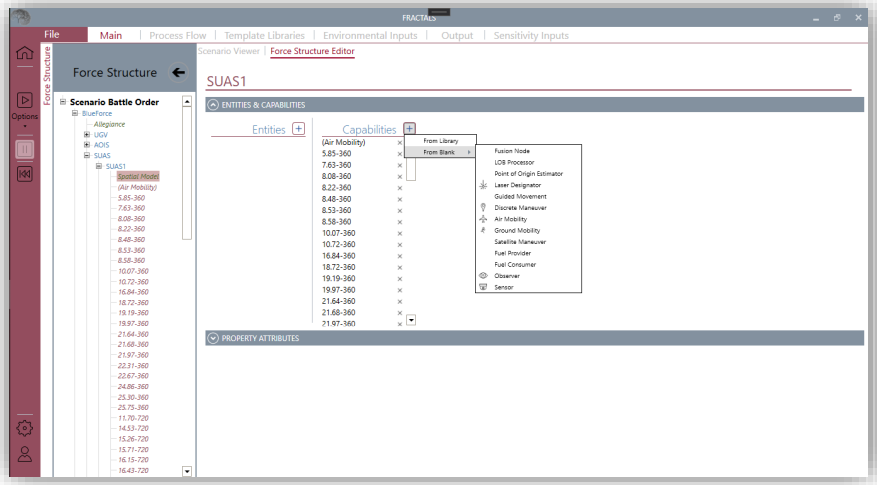
The Framework for Capability-based Tactical Analysis Libraries and Simulations (FRACTALS) is a DAC developed composable, component-level, framework designed to enable rapid development of tactical-level, event driven simulations to assess the performance and impact of complex, integrated systems.

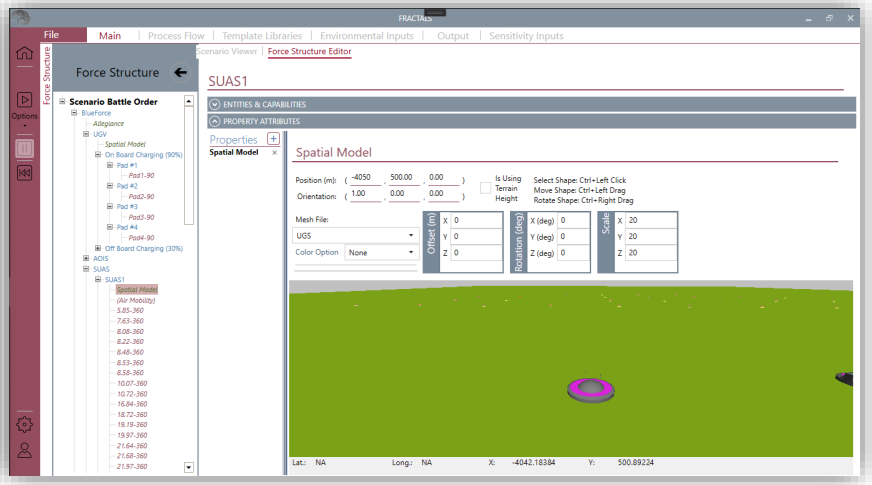
FRACTALS + TIPS: An ISR tool for item/system-level tactical performance analyses.

ISR: Intelligence Surveillance Reconnaissance
PED: Processing, Exploitation, and Dissemination
EO/IR: Electro-optical/Infrared
SIGINT: Signals Intelligence

FRACTALS - SYSTEM ARCHITECTURE

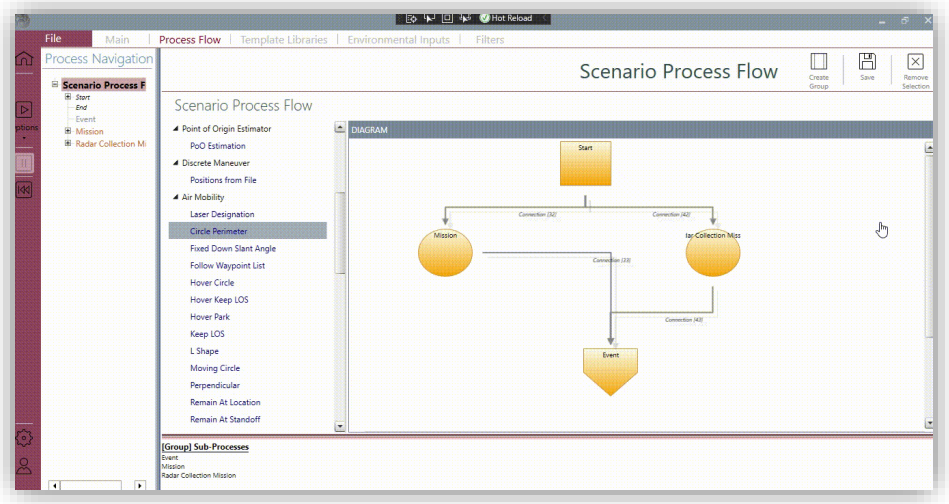
- **FRACTALS provides an API for building vignette-based event-driven simulations.**
- **Systems constructed with generic pieces:**
  - Entity – Named object (system or component) with attached sub-entities/capabilities/properties
  - Capability - encapsulated module of data, algorithms, and/or behaviors for performing a task/function (e.g., movement, sensing, communication)
  - Property - descriptive data, tags, and/or state attached to Entities or Capabilities (e.g., position, velocity, signature)
- **Developers implement capability modules and properties to build the functionality for a simulation configuration.**





API: Application Programming Interface

FRACTALS - PROCESS FLOW

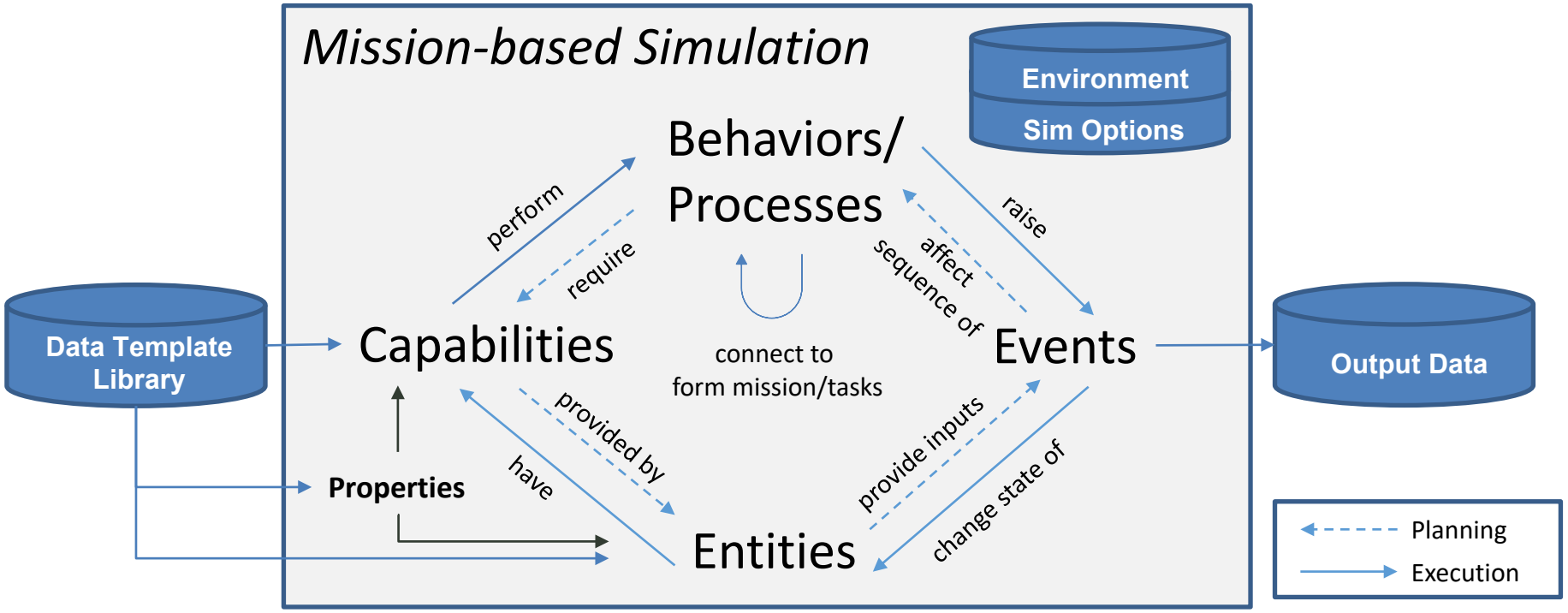


- **The Process Flow allows the user to construct dynamic execution paths in the simulation using visual coding.**
- **Create and connect process "blocks" to define the sequence of events in a vignette/scenario. Types include:**
  - Mission/Behavior – assigns entity or capability to perform a time-dependent task
  - Group – groups a set of process templates to enable dependent execution conditions
  - Group Task – a task that is defined as a group block consisting of multiple processes/behaviors
  - Procedure – changes state of object/executes a methodology to produce an event
  - Condition – applies filters on an incoming object to determine execution path
  - Event – sets "listening" event on capability and passes execution if event occurs
- **Connections between blocks allow data transformations in order to pair block outputs to block inputs that may be independently developed and not directly integrated.**

Process flows allow interactions between independent capabilities.

FRACTALS - SUMMARY

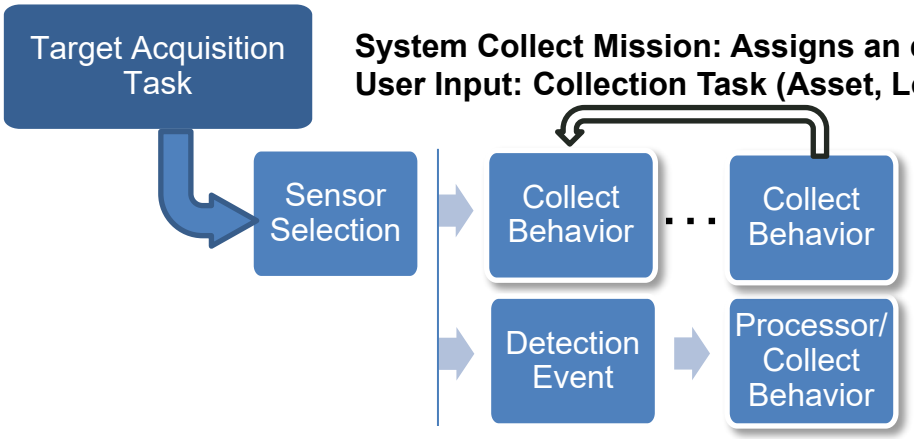
- **FRACTALS provides an extendable framework to build mission-based, tactical-level simulations.**
- **Newly developed capabilities using the FRACTALS API are automatically integrated into the user interface for use in scenarios.**



FRACTALS: An integrated and extensible tactical-level simulation framework for materiel performance analysis.

SENSOR SYSTEM MODELING

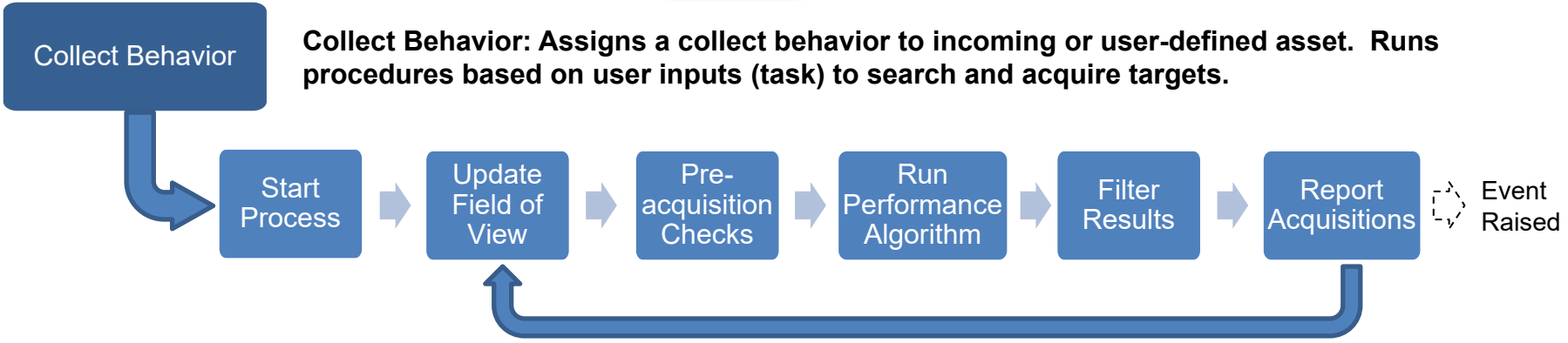
- **Sensor systems are modeled as groups of Sensing Capabilities.**
- **Behaviors are used to control the target acquisition timeline of the "system" given a searchable object/location, a search pattern and/or sequence, and an optional target.**
- **Sensor Observation "Processor" capabilities can handle complex logic for processing acquisition events (e.g., Radar tracking revisits)**



System Collect Mission: Assigns an entity (system) to perform collect task.
User Input: Collection Task (Asset, Location, Objective)

Sensor Selection (Procedure): Chooses a single "Sensing Capability" to collect task passed as input from system behavior.

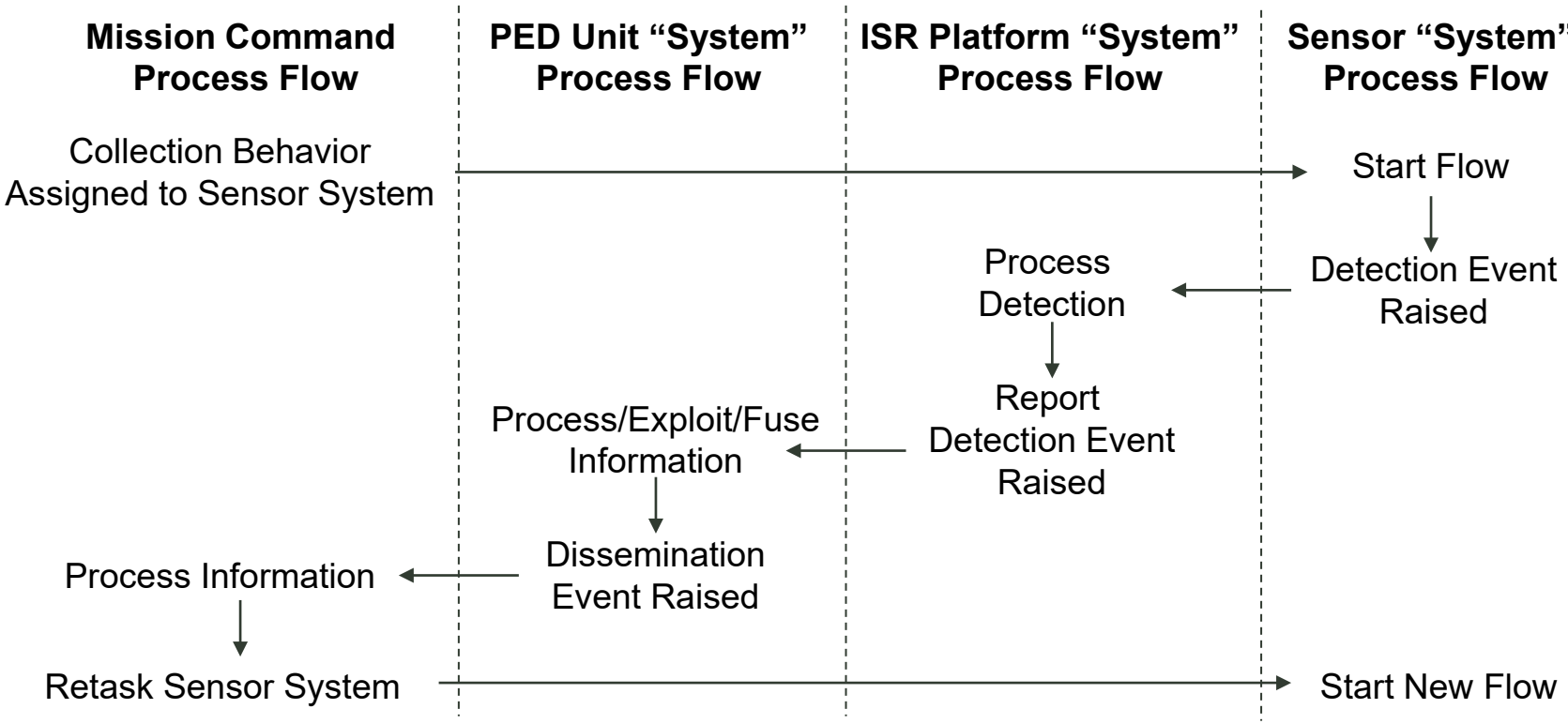
Detection Event: Sets a listener for the detection event for the capability chosen in the sensor selection procedure.



Collect Behavior: Assigns a collect behavior to incoming or user-defined asset. Runs procedures based on user inputs (task) to search and acquire targets.

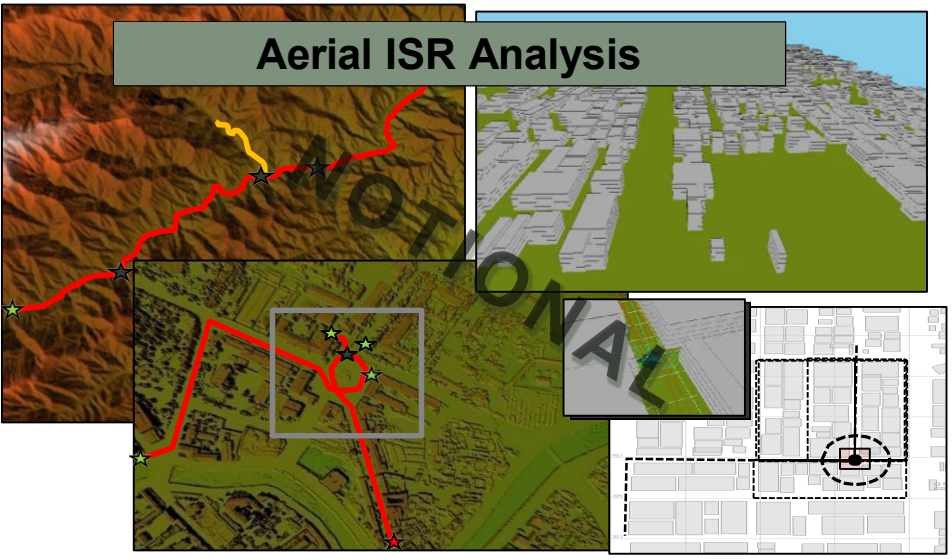
INTEL PED / DATA FUSION MODELING

- **Intelligence PED modeling is achieved through the process flow architecture, where ISR capabilities route information from entity to entity and perform PED.**
  - Can represent on/off-board processing and the data links to other entities (e.g., PED node).
  - Flexibility to design PED "threads" at varying levels of detail.



Users can construct custom intelligence process flows.

SAMPLE ANALYSES



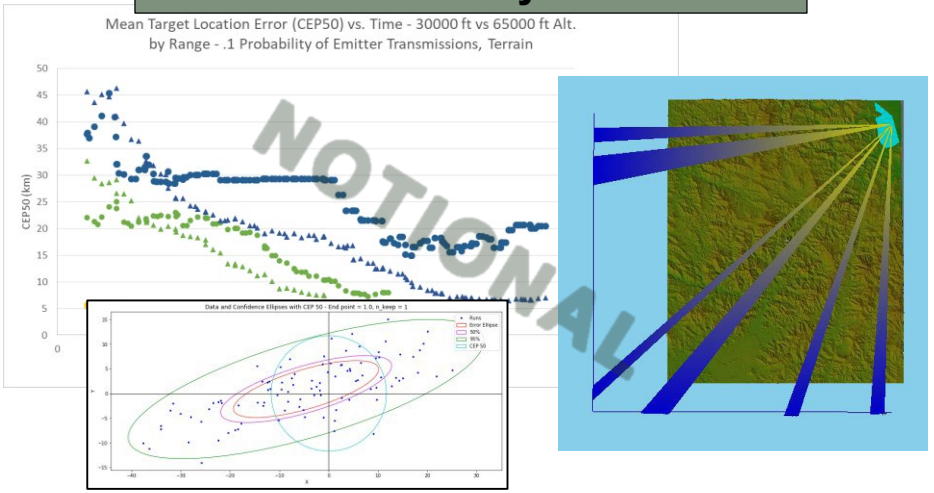
Aerial ISR Imaging Analysis

- Effectiveness of multiple FMV vs. single FMV capability in tracking and search detection/identification over a designated time period
- Impact of static vs. dynamic TTPs in tracking
- Impact of terrain artifacts on detection & tracking capabilities
- Assessment of hovering vs. fixed wing acquisition performance in tactical urban vignettes
- Impact of gimbal vs. fixed sensor performance in executing tactical urban operations

Target Location Error Sensitivity Analysis

- SIGINT Line of Bearing Sensitivity
  - Modeling the sensitivity of TLE over time with varying lines of bearing using an Angle of Arrival methodology.
  - Varied vignette with multiple flight path lengths, standoff, altitude, and emitter transmission intervals
- Radar – Point of Origin
  - Modeling the estimation of a munition point of origin based on Radar track estimates.
  - Enhanced Radar methodology with multiple scan modes and track revisit rate.
  - Polynomial Fit and Reverse Kalman Filter Point of Origin methodologies developed.

TLE Analysis

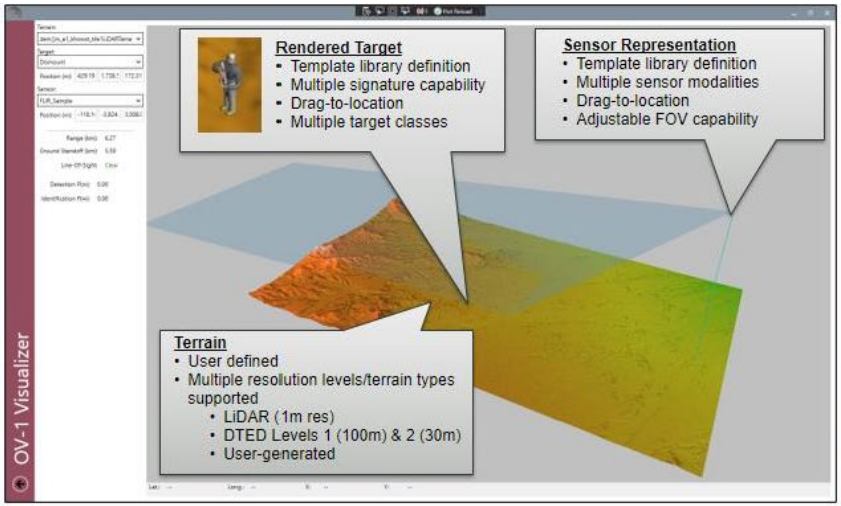


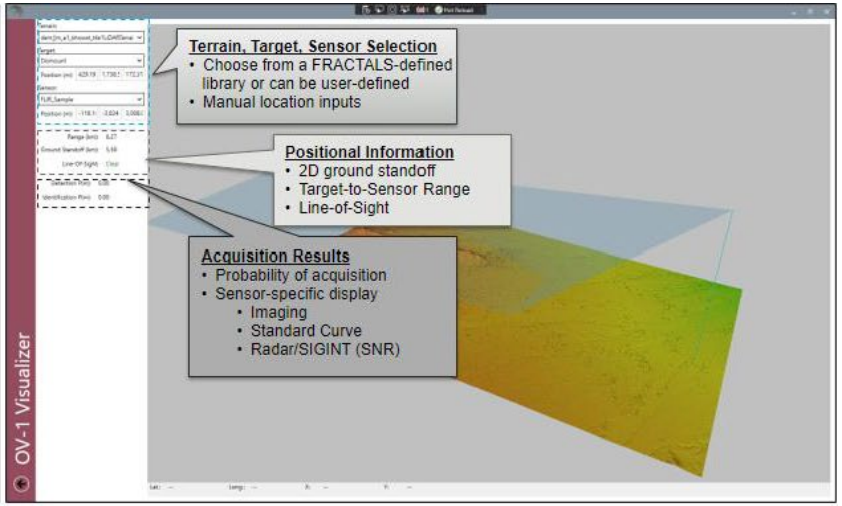
FMV: Full Motion Video
TTP: Tactics, Techniques, and Procedures
TLE: Target Location Error

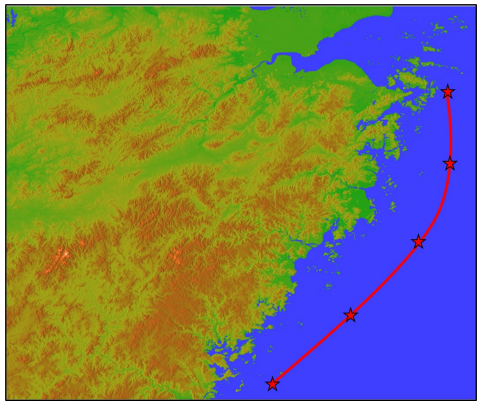
DYNAMIC OV-1 VISUALIZATION & COVERAGE ANALYSIS

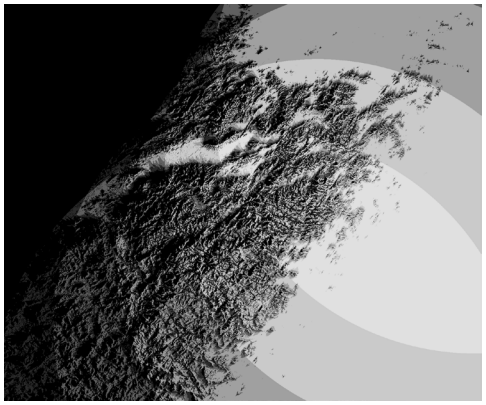
Purpose: Quickly & dynamically visualize effects of platform and sensor trades to more effectively communicate impacts to those without extensive expertise. Generate images of high-level sensor coverage information.

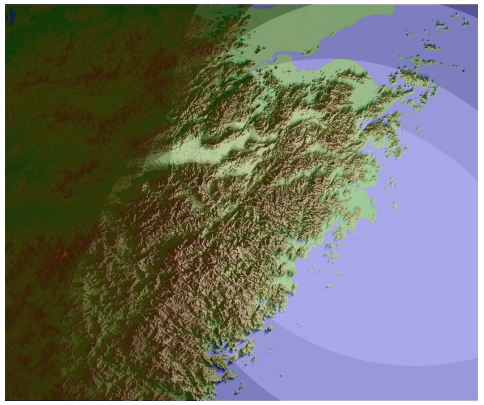
Impact: Gives analysts a real time visualization tool to brief Senior Leaders that is more impactful than graphs and charts.





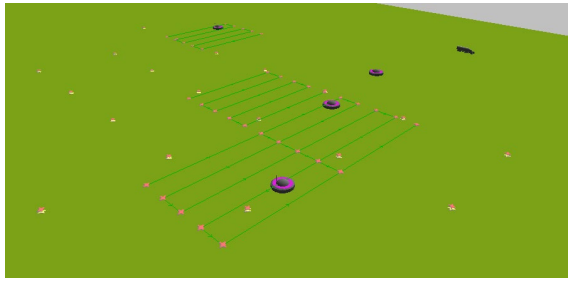


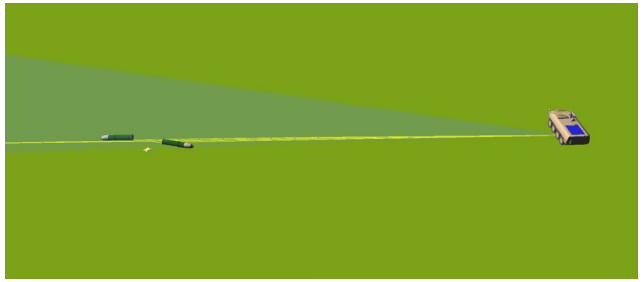


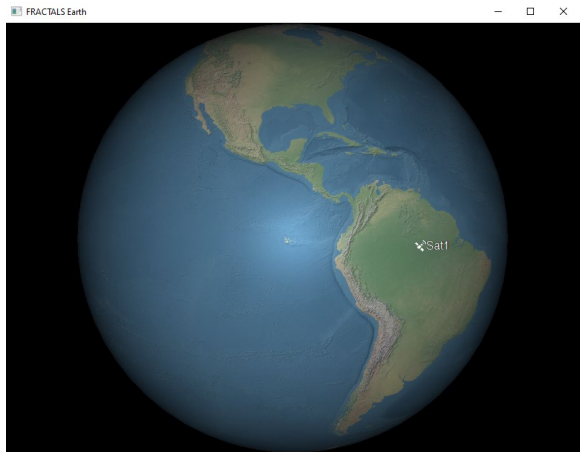


CURRENT/FUTURE DEVELOPMENT & ANALYSES

- Power & Energy
  - Sensitivity analysis on energy consumption rates, recharge rates, and number of consumers/provider stations
  - Assessment of varying number of small UAS flying area coverage routes
  - Impact of recharge at a standoff
- Active Protection Systems (APS)
  - End to end modeling of APS hard kill solutions including threat trajectory, Radar measurements and tracking, fire control systems, launcher, and countermeasure fly out/result
  - Assessment of potential Radars and several fire control system/countermeasures
- Space ISR Analysis
  - Full globe visualization
  - Satellite vehicle movement
  - Space-based sensor coverage and performance







IMPACT

FRACTALS enables rapid development of multi-function, integrated systems for quick-turn performance analysis.

- Enables external development of capabilities
- Supports improved long-term development
- Supports better scalability and expansion
- Supports user development of new system capabilities
- Easy to use interface and visualization of systems

TIPS encapsulates DAC's validated ISR models for reuse and provides methodologies to simulate the end-to-end Intelligence Process.

- Code library available for external applications
- Possible use as data generator/item-level performance model
- Greater flexibility in creating complex ISR systems and behaviors without code changes