Two gaps that need to be filled in order to trust AI in complex battle scenarios

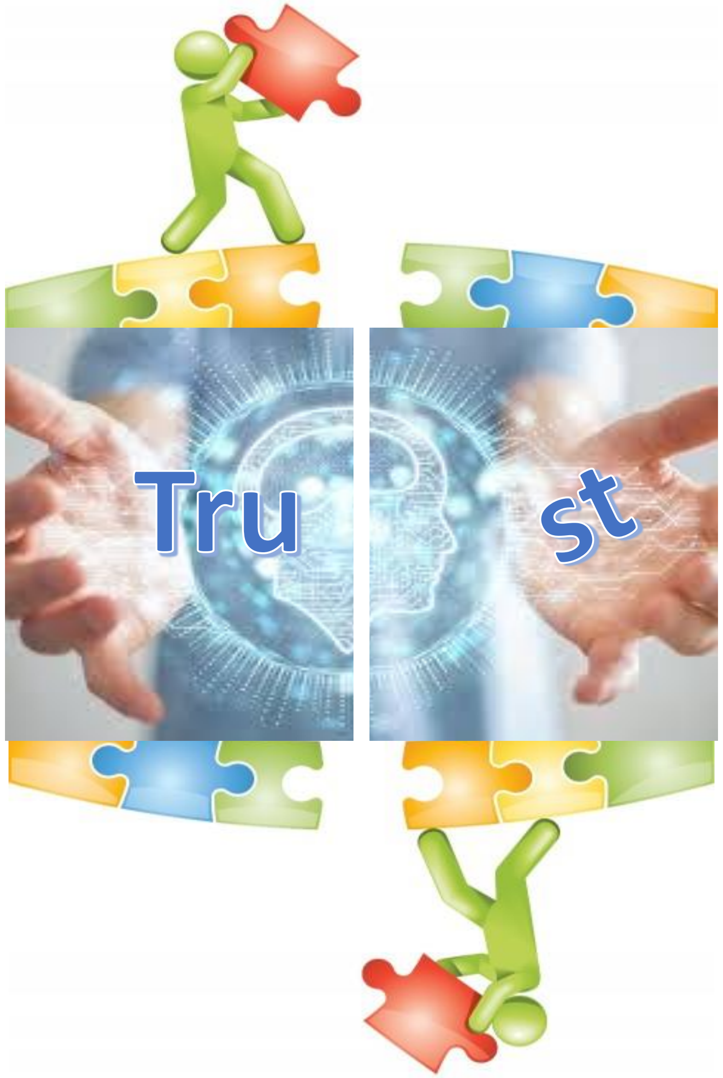
Presented for the Acquisition Research Symposium, May 11-12, 2022

(Panel) Designing and Deploying Artificial Intelligence to Improve Performance

Presented by Bruce Nagy

bruce.m.nagy.civ@us.navy.mil

NAVAIR, NAWCWD China Lake



Introduction

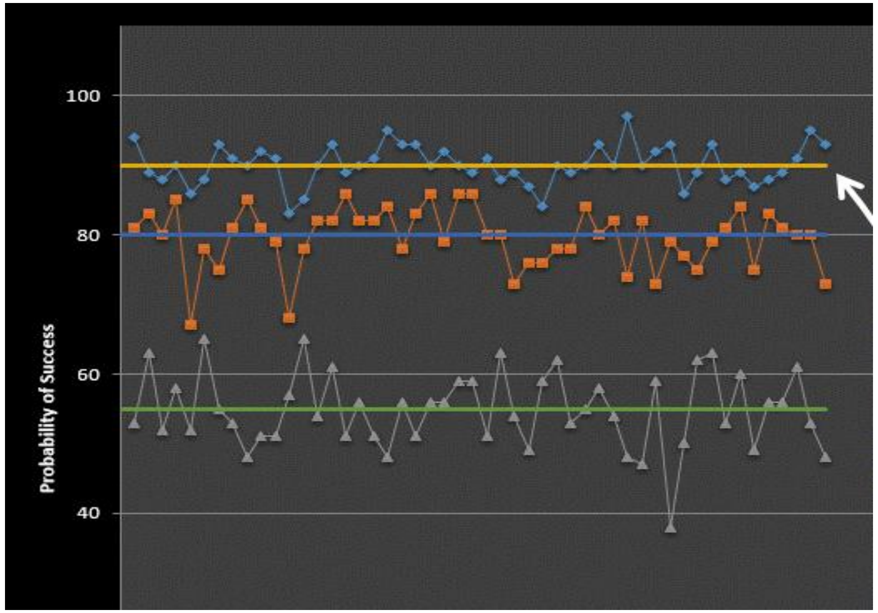
- In human terms, **trust is earned**
- Should it **be different for AI** when providing recommendations?
- Paper/presentation discusses filling two gaps for an AI-based **Course of Action (COA) recommendation algorithm** (CRA)
- It introduces a **nine-stage process** (NSP) divided into **three phases** for a CRA to earn trust with its human users **through dataset development**
- The NSP is dependent on a concept called **Event-Verb-Verb** (EVE) and **EVE Segments to support dataset development**
- **EVE Segments** allow CRAs to be trained with a **combination of theory and practice** to provide more practical and accurate recommendations.

Course of Action (COA) recommendation algorithm (CRA) Tasking in Wargames (**Gap 1 – Dataset Development**)

- CRA needs to be **developed from a wargaming environment** to capitalize on a "treasure trove" of move-to-counter-move knowledge and possibilities, such as:
  1. **human factors** that can affect outcomes,
  2. **unanticipated/surprise** moves changing battle results,
  3. **multidomain scenarios**, where joint and coalition forces are integrated to achieve a common goal, and
  4. the ability to accurately **interpret** various qualities of **intelligence/sources**.
- CRA needs to learn how to **unravel battle complexity**, including uncovering and managing "unknowns," and still be able to determine an optimal strategy/tactical response

Course of Action (COA) recommendation algorithm (CRA) Tasking in Operational Tests (**Gap 2 – Dataset Development**)

- CRA **working with actual "live" operational** testing of new technology products being developed/acquired by Department of Defense (DoD) programs.
- CRA **learn from firsthand experience** what products can and cannot do.
- Use learning/data can to **refine the moves and countermoves** discussed during the wargaming exercises. (This also ensures accuracy in the recommendation.)



NAVAIR

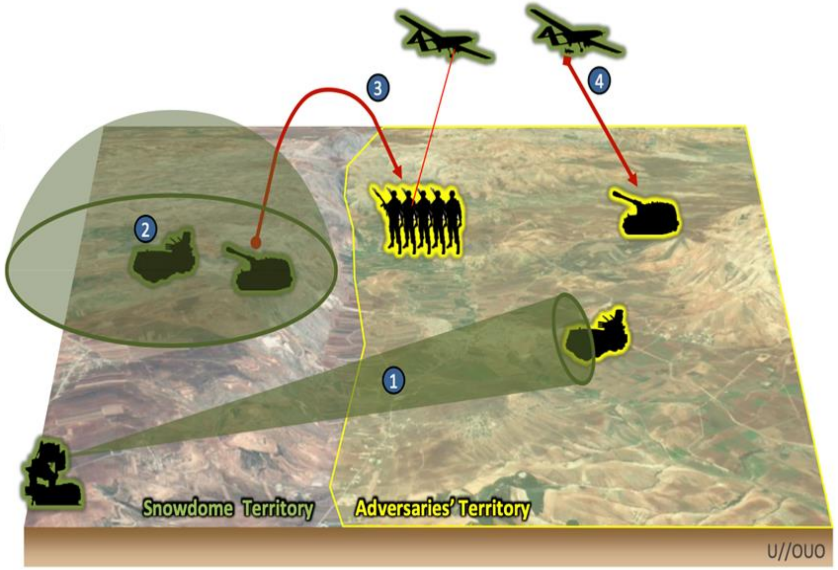
Battle Surprise – was it in the dataset?

- In a 1955 news conference, President Dwight D. Eisenhower stated, *“every war is **going to astonish you** in the way it occurred, and in the way it is carried out.”* (Eisenhower Library 2022).
- From the Center for Strategic and International Studies (CSIS) article, Cancian defined surprise as “***when events occur that so contravene the victim’s expectations that opponents gain a major advantage***.”
- How is this problem addressed in AI? Ensure the datasets used to train the AI system accurately reflect the deployed operational state!
  - **Gap 1: The need for extensive wargaming and**
  - **Gap 2: The need for operational testing before deployment.**
- Filling these two gaps are **not optional, they are required** to ensure trust in the CRA.

Learning from History – When Developing a CRA

Lessons Learned:
Pearl Harbor

Lessons Learned:
Battle of Midway



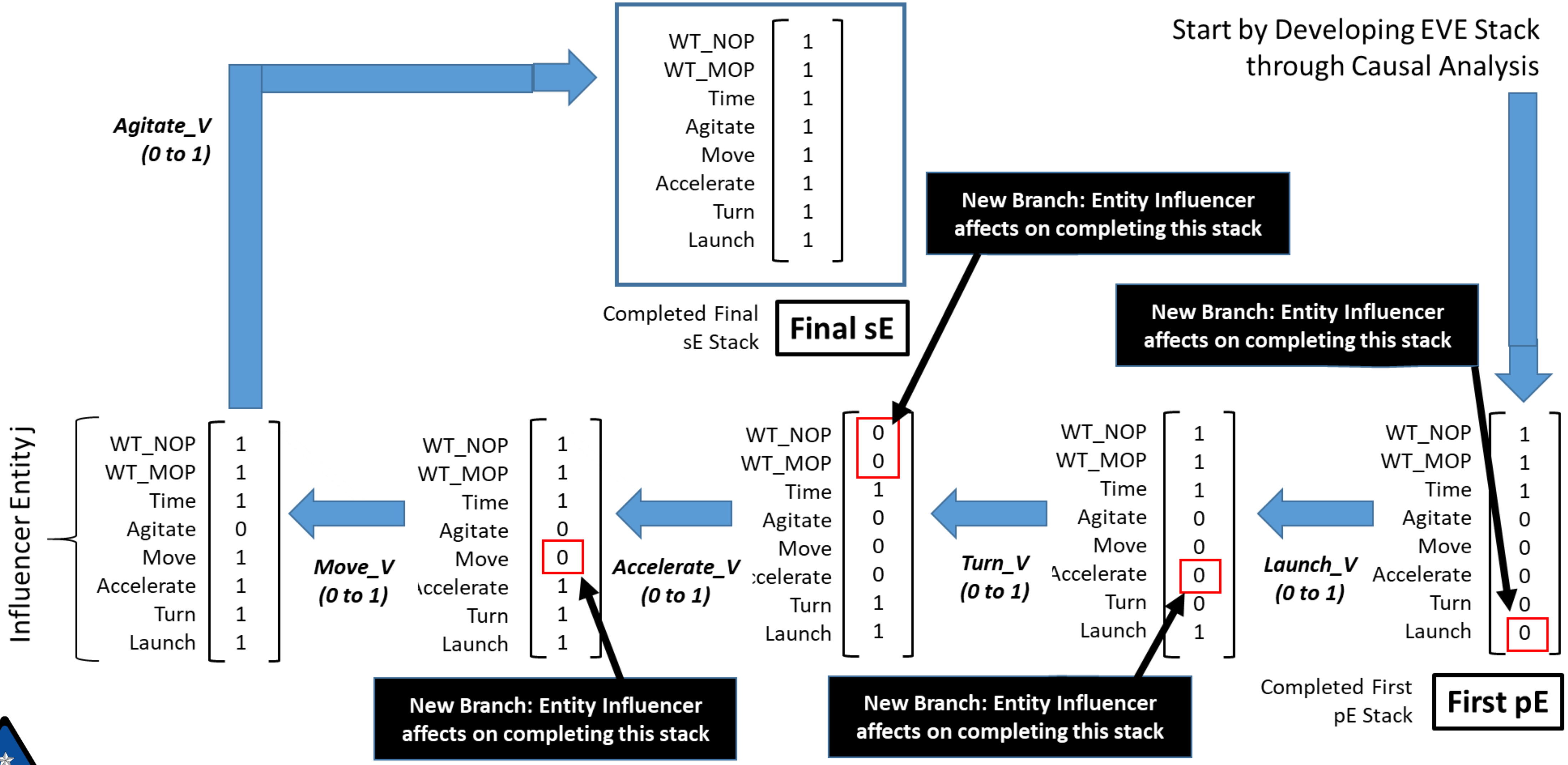
Lessons Learned:
Nagorno-Karabakh War

Chinese entry into the Korean War, **North Vietnamese** offensive during the Tet holiday, **Egyptian and Syrian** attacks on Israel in 1973, the fall of the Shah in 1979, the fall of the berlin wall in 1989, terrorist attacks on September 11, 2001 and the tenacity of Ukrainian civilians to stand up to a Russian attack on their homeland.

Developing CRA using Lessons Learned from History

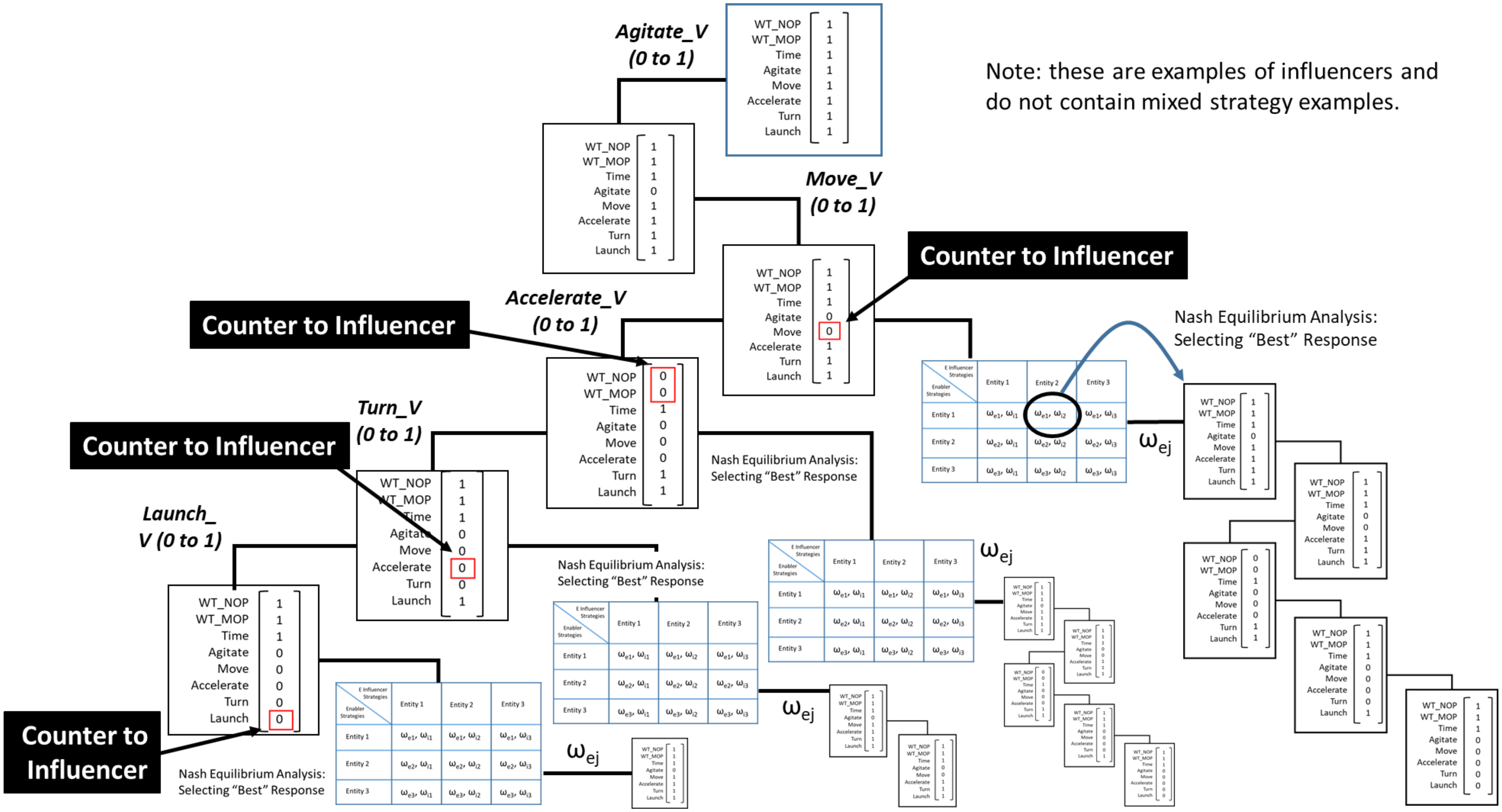
- **Explainable AI may not be enough** when significant change is needed
  - December 7, 1941 attack on **Pearl Harbor**
    - Could a CRA have predicted the December 7, 1941 attack on Pearl Harbor? If so, would leadership have trusted the prediction?
    - Cancian points out that the **attack on Pearl Harbor was predicted** the problem however was a **lack of trust in those predictions**. Past History: Japanese attacked the Russian's Port Arthur in China about half a decade ago
    - Without trust, a military commander is not likely to commit a sizeable number of military resources based on a machine's recommendation?
  - Consider the **Battle of Midway**
    - Japanese had superior forces, more experienced pilots, better aircraft, and an element of surprise
    - Japanese did not account for the **Americans breaking their code**, but they didn't account for **American bravery**
- **Trust to overcome hubris** may be the best approach
  - Can the hubris make a CRA recommendation even harder to accept?
  - Battle of **Nagorno-Karabakh War**
- Avoid designing a **Course of Action (COA) recommendation algorithm (**CRA) to earn limited trust

The Power of Event-Verb-Events (EVEs) -Identifying weaknesses in strategy/tactics in order to **Build the "Right" Algorithm Dataset**



The Power of EVEs -Identifying counter moves and resiliency solutions to Continually **Build the "Right"** **Algorithm Dataset**

Note: these are examples of influencers and do not contain mixed strategy examples.



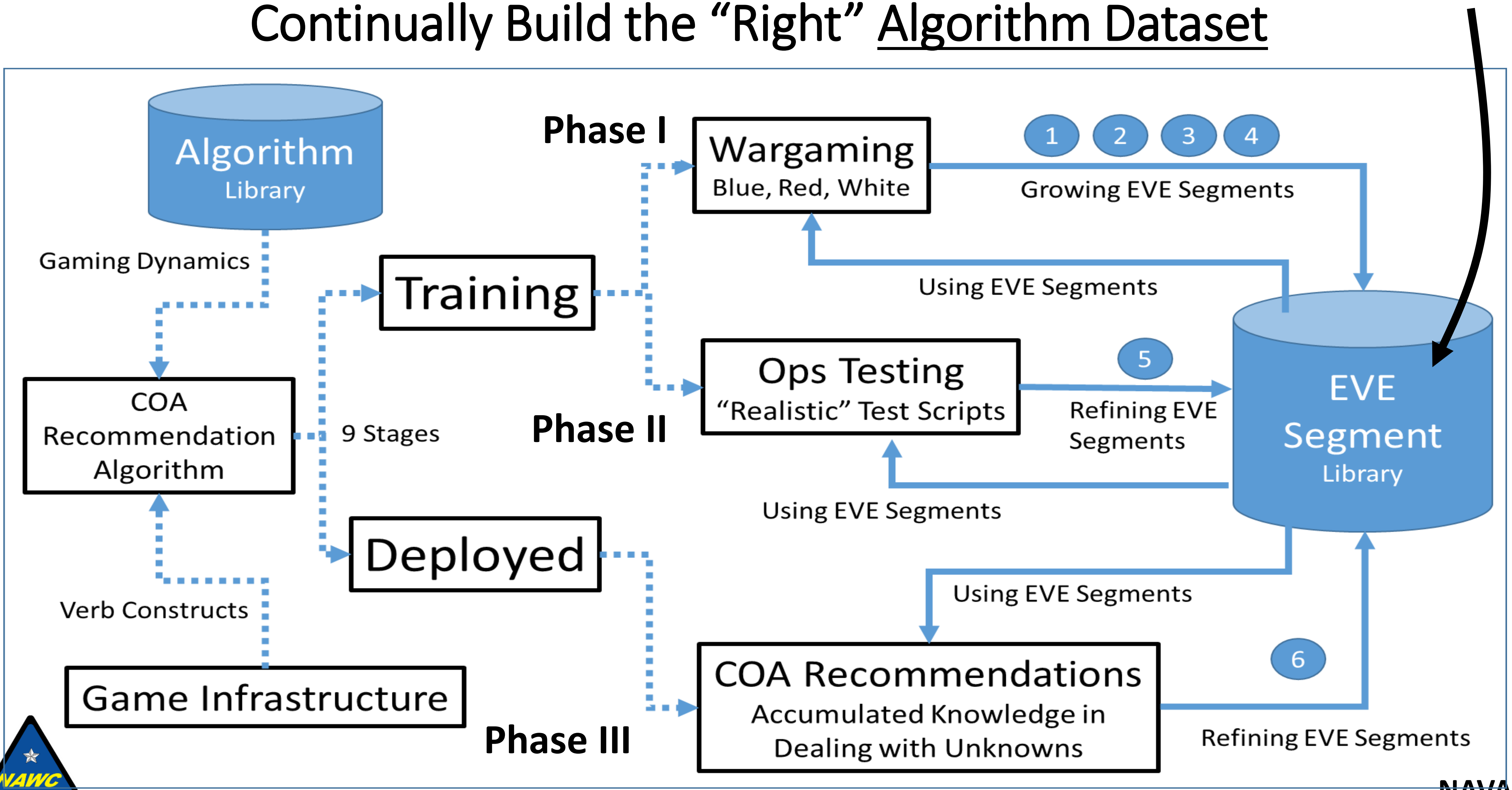
EVEs related to AI in Complex Battle Scenarios:

- For Event-Verb-Event (EVE) chains associated with AI systems, battle complexity is defined as a situation that can be described by a series of events, i.e., EVE chains, caused by actions between opposing participants, where the outcomes can be significantly affected by factors categorized as: (1) “known-knowns” (facts), (2) “known-unknowns” (assumptions) (3) “unknown-knowns” (absent data) and (4) “unknown-unknowns” (surprises).
  - **“Known-knowns” (facts)** – AI/ML Dataset - These are EVE chains from data collected from wargames and operational tests.
  - **“Known-unknowns” (assumptions)** – AI/ML allowed variations from Dataset. These are assumed variations in EVE chains from data collected from wargames and operational tests.
  - **“Unknown-knowns” (absent-data)** – AI/ML Sparse and Missing Data associated with Dataset. These are missing state variables in EVE chains.
  - **“Unknown-unknowns” (surprises)** – AI/ML unbound variations from Dataset. These are EVE chains that have not been identified in any wargame or operational test.
- The Nine Stage Process (NSP) will describe how these EVE chains are addressed using generalization (Stage 9, Phase III of the NSP approach).

NAWC Weapons Division NAVAL AIR WARFARE CENTER

NAVAIR

Three Phase and Nine Stage Process (NSP) Overview – Designed to Continually Build the "Right" Algorithm Dataset



NAWC Weapons Division NAVAL AIR WARFARE CENTER

NAVAIR

Developing CRA Segments 1 to 3 - **Continually Building the "Right" Algorithm Dataset**

Phase I Wargaming - Segment 1:

- Create Verb Infrastructure
  - EVE Ontology
  - Verb Table
  - Verb Binary Codes
  - Verb Hex Code
- Create Algorithm Library
  - Geometry for Global Movement Dynamics
  - Optimization using Learning Rate of Attributes
  - Statistical Significance Analysis
- Create EVE Segment Library (with Sections)
  - Manageable Obstructions
  - Unmanageable Obstructions
  - Enabler Mission Actions
  - Influencer Actions
  - Influencer Counter Actions

★ Establishing EVE Segment Library

Phase I Wargaming - Segment 2 (for Blue as Enabler):

- Define Mission Constraints
  - Blue Mission Criteria
  - Blue Performance Area
  - Blue Environmental Influencers
    - Immovable Obstacles
    - Moveable Obstacles
    - Weather and Other Conditions
  - Blue Entities
    - Blue Actual Performance Specs
    - Blue Allie Estimated Performance Specs
  - Red Influencer Entities
    - Red Estimated Performance Specs
    - Red Allie Estimated Performance Specs

Phase I Wargaming - Segment 3 (for Blue as Enabler):

- Develop Ideal EVE Chain
  - Movement Dynamics
    - Performance Area
    - Immovable Obstacle
    - Environmental Conditions
  - Mission Achievement
    - Tree Trunk and Branches
      - Verb Stack
      - EVE Stack (Values)
      - Binary and Hex Code EVE Stack
  - EVE Segment Library by Move, Move (Results)
    - Store Optimal Strategy
    - Store Statistical Measured Results

1 Growing EVE Segment Library

NAWC Weapons Division NAVAL AIR WARFARE CENTER

NAVAIR

Developing CRA Segments 4 and 5 - **Continually Building the "Right" <u>Algorithm Dataset</u>**

Phase I Wargaming - Segment 4
(for Blue as Enabler):
- Develop Non-Ideal (Challenged) EVE Chain
  - Movement Dynamics
    - Performance Area
    - Environmental Influencers
      - Immovable Obstacles
      - Moveable Obstacles
      - Weather and Other Conditions
    - Entity Influencers
      - Blue and Allie Actual Performance Specs
      - Red and Allie Estimated Performance Specs
  - Mission Achievement
    - Tree Trunk and Branches
      - Verb Stack
      - EVE Stack (Values)
      - Binary and Hex Code EVE Stack
    - Counter Moves
      - Verb Stack
      - EVE Stack (Values)
      - Binary and Hex Code EVE Stack
  - EVE Segment Library by Move and Counter Move
    - Store Optimal Strategy
    - Store Statistical Measured Results

2 Growing EVE Segment Library

Phase I Wargaming - Segment 5
(for Red as Enabler):
- Repeat Segment 2 to 4 for Red Team
  - Segment 2' (for Red as Enabler):
    - Define Mission Constraints
      - Red Mission Criteria
      - Red Performance Area
      - Red Environmental Influencers
      - Red Entities
      - Blue Influencer Entities
  - Segment 3' (for Red as Enabler):
    - Develop Ideal EVE Chain
      - Movement Dynamics
      - Mission Achievement
      - EVE Segment Library by Move and Counter Move
  - Segment 4' (for Red as Enabler):
    - Develop Non-Ideal (Challenged) EVE Chain
      - Movement Dynamics
      - Mission Achievement
      - EVE Segment Library by Move and Counter Move

3 Growing EVE Segment Library

NAWC Weapons Division NAVAL AIR WARFARE CENTER

NAVAIR

Developing CRA Segments 6 and 7 - **Continually Building the "Right" Algorithm Dataset**

Phase I Wargaming - Segment 6:

- Adjudicate War Game as White Cell
  - Run Monte Carlo Wargame with Existing Assumptions of Blue and Red EVE chains against each other
    - Blue EVE Chain using Assumptions based on Red Intel about Capability
    - Red EVE Chain using Assumptions based on Blue Intel about Capability
    - Statistics by Segment
      - Store for Enabler (Blue) with Influencer (Red) Assumptions
      - Store for Enabler (Red) with Influencer (Blue) Assumptions
  - Run Monte Carlo Wargame with "Truth" of Blue and Red EVE chains against each other
    - Blue EVE Chain using Red "Truth" about Capability
    - Red EVE Chain using Blue "Truth" about Capability
    - Store Statistics by Segment

Phase I Wargaming - Segment 7:

- Review Lessons Learned
  - Comparison of Delta's Assumption vs Truth
  - Connect Delta's to Statistical Results (Answers Why)
  - Develop Optimal Solutions based on "Truth" from both sides (EVE Segments with Statistical Results)
  - Store in EVE segment database by Move, Counter Move (Results from Blue and Red)
    - Optimal Strategy
    - Non-Optimal Strategy
    - Statistical Measured Improvement

4 Growing EVE Segment Library

Developing CRA Segments 8 and 9 - **Continually Building the "Right" <u>Algorithm Dataset</u>**

Phase II T&E - Segment 8:

- Test Script Generator
  - Import Mission Requirements for Test and other Segment 2 Data
  - Mix and Match EVE Segments to create
    - Optimal Solution: (1) EVE Tree with Causal Why, and (2) Why Statistically based on "Truth" of Influencer
    - Nominal Solution: (1) EVE Tree with Causal Why, and (2) Why Statistically based on "Truth" of Influencer
    - Stressed Solution: (1) EVE Tree with Causal Why, and (2) Why Statistically based on "Truth" of Influencer
  - Based on testing, modify EVE segments used to support measured results, including variations in statistics
  - Store Data Changes from Test Results in EVE segment database by Move, Counter Move (Results from Blue and Red)

5 Refining EVE Segment Library

Phase III Deployed Operations - Segment 9:

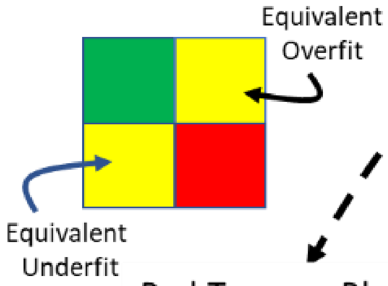
- COA Recommendation Engine
  - Import Mission Parameters and other Segment 2 Data
  - Allow User Preference
  - Mix and Match EVE Segments to create a Pareto Chart associated with BRE Matrix
    - Solution given Assumed Intel Truth and Use Preference
    - Solution given Variations in Intel Assumptions based on Wargaming
  - Provide Solution that encompasses as many points on the "Green" segment of the Matrix
    - Not optimal for a single point
    - Best compromised solution for encompassed points
  - Store Data Changes from "Live" Operational Results in EVE segment database by Move, Counter Move (Results from Blue and Red)

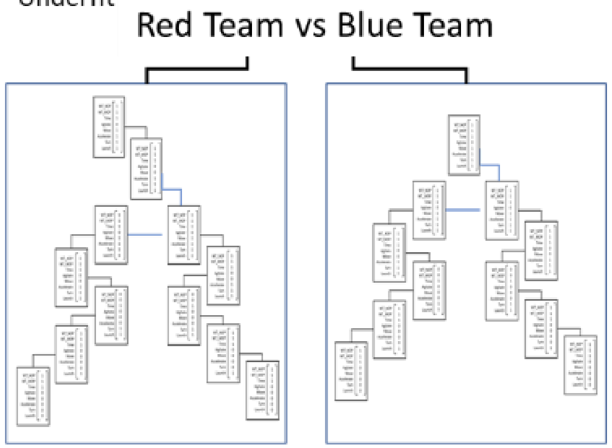
6 Refining EVE Segment Library

NAWC Weapons Division NAVAL AIR WARFARE CENTER

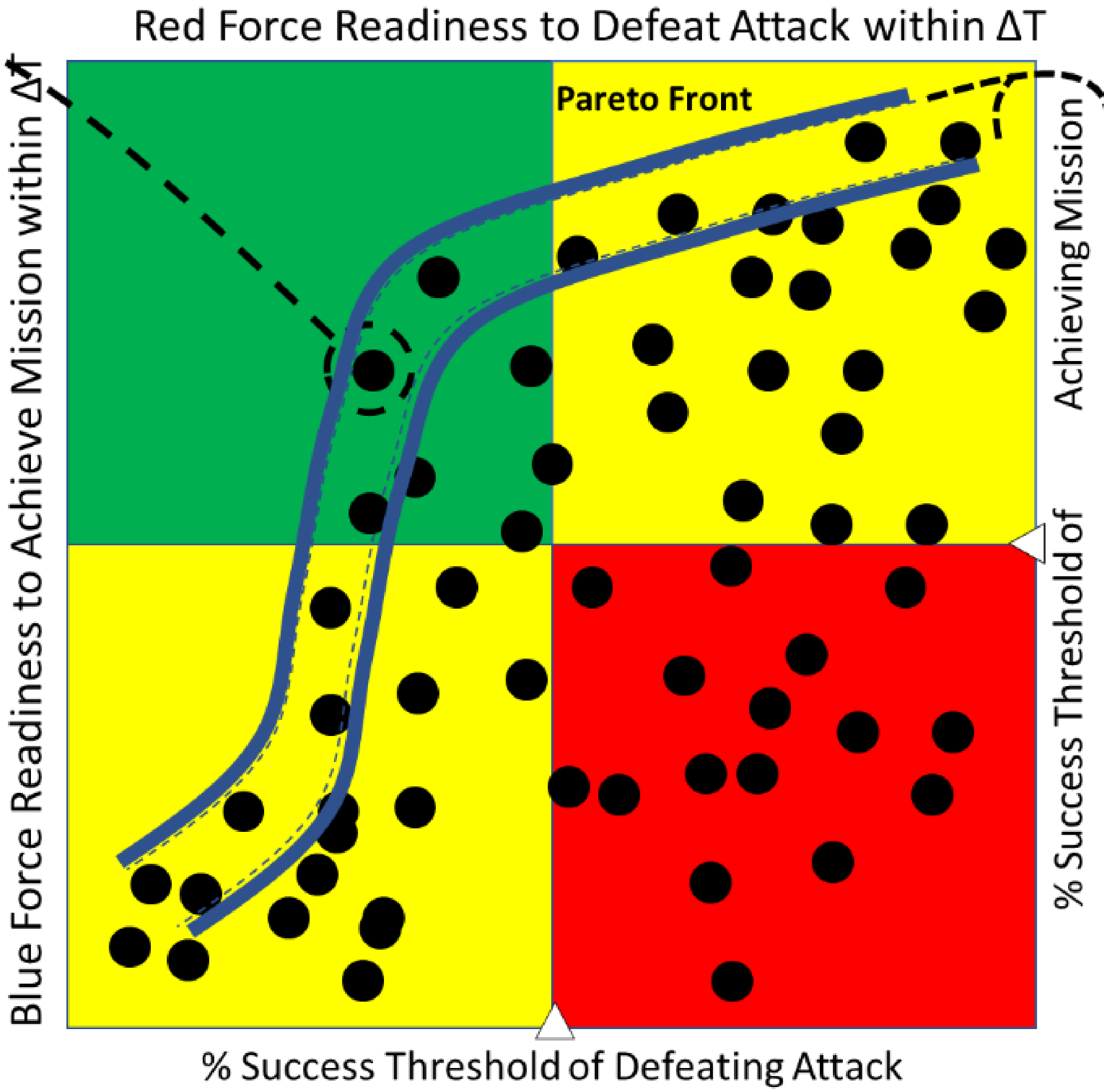
NAVAIR

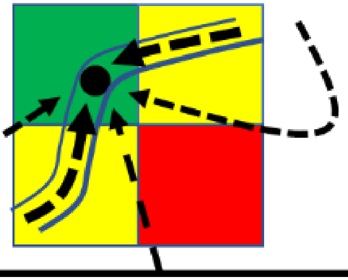
Finding EVE Tree that Optimizes Ability to Succeed Independent of Opponent Strategy – to Use **the "Right"** **Algorithm Dataset**





Blue Force (Enabler) EVE tree is limited because of Intel Inaccuracies about Red Force (Influencer) – thereby its assumptions about Influencer EVE tree and countermoves have inherent errors – *Blue perspective limitation!*





- The White Cell was trained to know Truth given many variations of Intel inaccuracies
- Research to investigate creation of a single Offensive EVE tree that is optimized to ensure success rate ≥ 80% even if intel is inaccurate (within range of Pareto Front) about the opponent's defensive intent, actions and/or capabilities

WIFM Human Factor to Address – **Ensure People Support the Development of the "Right" Algorithm Dataset**

- **Professional wargamers:**
  - Automate their existing tool suite – moves and countermoves can be more easily entered and analyzed with significantly greater statistical precision.
  - Automate adjudication process – statistical models vs roll of the dice analysis
  - Provide realtime "what-if" analysis of wargame strategies
- **Operational test engineers:**
  - Auto-generate test scripts that more accurate replicate deployed experiences
  - Enhance statistical analysis of test results to find stress areas of product under test
  - Automate decision support for constrained test scenarios that are challenged with creating "realistic" battle engagement test scripts and real/synthetic environments for autonomous systems, including manned and unmanned teaming.

Conclusions - AI Architype for Earning Trust – Creating and Using **the "Right"** Algorithm Dataset

1. **(Gap 1 – Dataset Development)** Have the AI learn from Subject Matter Experts (SMEs), where its learning can be continually tested/validated, thereby proving performance
2. **(Gap 2 – Dataset Development)** Have the AI be involved with "real" technology, learning from firsthand experience what systems can and cannot do, where its learning can be continually tested/validated, thereby proving performance.
3. **(WIFM – Dataset Development)** A final key aspect to using this architype is ensuring that any human involved with the training of the AI receive value, i.e., his or her motivation factor is also filled during the process.