A Trustable Autonomous Systems Lifecycle

Howard Reubenstein, BAE Systems Technology Solutions

Autonomy in Motion
Core Challenges

- Autonomous unmanned systems (AUS) will operate in contested, complex environments.
  - Future fight will be urban, close, and vertical.
  - Cross domain force protection, expedition, and situational awareness maneuvers with minimal C2 infrastructure.
- Warfighters require flexible AUS:
  - Tunable rules of engagement and autonomous decision making.
  - Learn behaviors to increase in-mission adaptability.
- Good AUS performance builds warfighter confidence and reduces cognitive load.
  - Autonomy reduces required bandwidth and security constraints.
  - Autonomy provides a force multiplier – operators of multiple AUS.

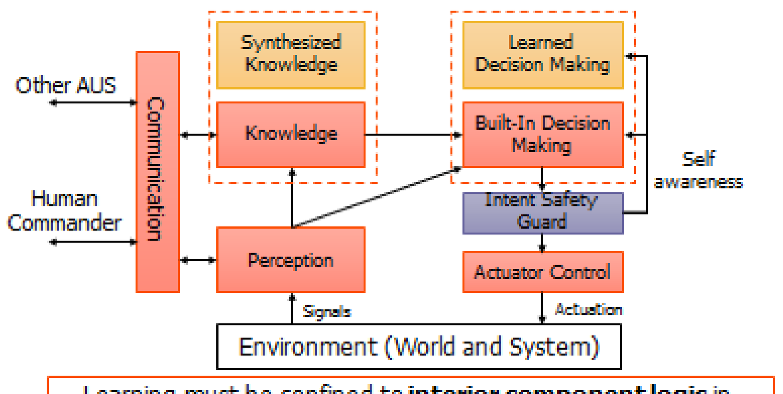
Trustable?
Obeys Rules of Engagement
Implements Commander's Intent

Autonomy in Motion
Establishing Trust

- Acquisitions lack methods to evaluate and certify AUS programs of record.
  - How do you know a learning algorithm "works?"
- Manufacturers need to integrate multiple autonomous system modules.
  - How do you know collections of learning-based, autonomous modules do not cause unstable interactions?
- Warfighters have to build trust with AUS in manned-unmanned teaming.
  - How much should a system adapt and learn, e.g. variable bounds on autonomy and engagement?
  - How should the commander/operator interact to encourage (or, discourage) the learning?

If an autonomous system does not have dynamic behavior guarantees, it will not be purchased, built, or used.

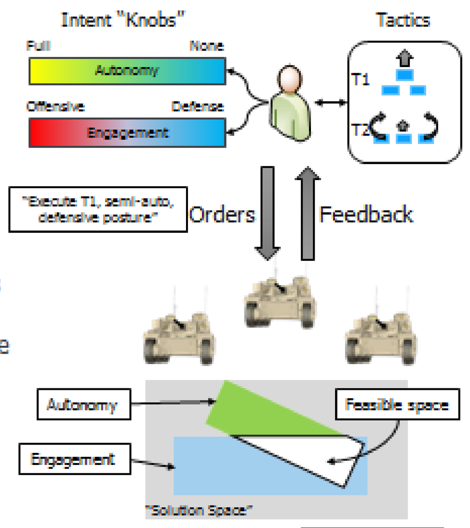
Sandboxed Autonomy
Notional Architecture



Learning must be confined to **interior component logic** in order to preserve system verification and validation arguments.

Sandboxed Autonomy
Incorporating the Human

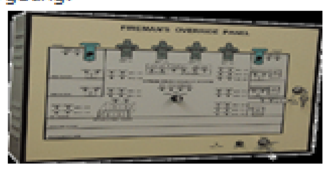
- Humans must be in the loop:
  - Tactical tasking
  - Intent specifications
  - Coaching and corrective action
- Balance autonomy, rules of engagement, & human interaction:
  - Too many restrictions and there is no (or, bad) learning
  - Commander intent must afford the AUS with enough expressivity to achieve tactical tasks



Intent Safety Guard
Vehicle Maneuvers Example

- Basic intent behaviors include:
  - Do not exceed 80MPH land speed
  - Do not accelerate towards solid obstructions
  - Maintain safe following distance to vehicle ahead
  - Do not pass on the right
- More complicated tactical maneuvers:
  - Move from point A to point B avoiding exposed environments.
  - Encircle red forces while avoiding blue force targeting.

The intent safety guard allows a wide range of behaviors but keeps the AUS in a safety envelope (potentially over-ridable).



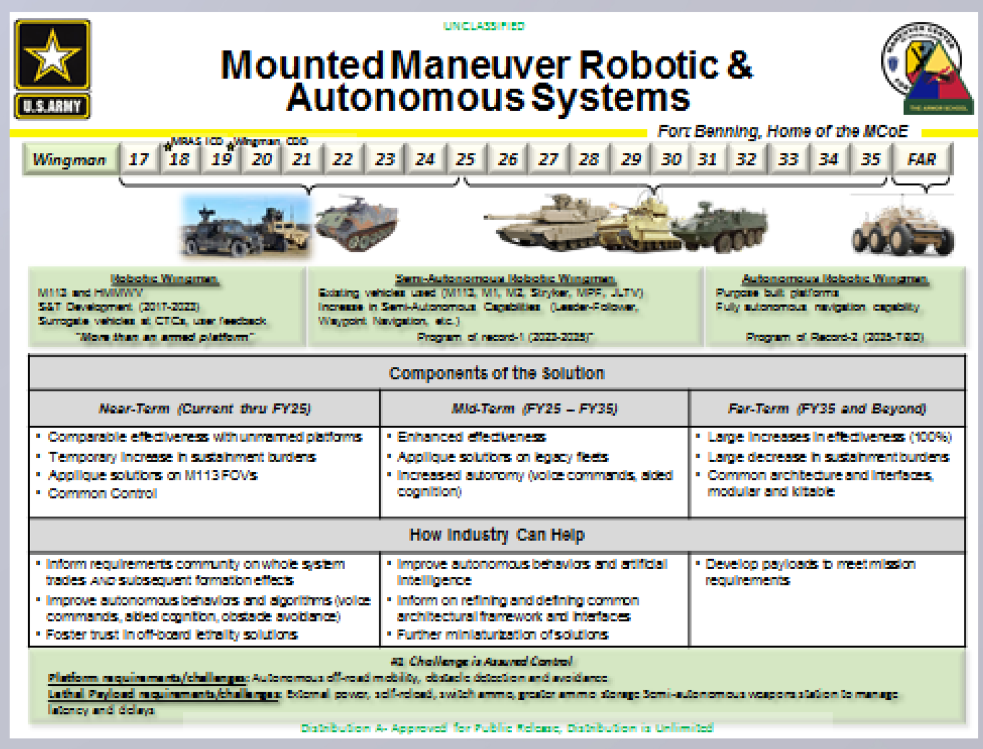
Software Intent Specification
Provides a Behavioral Envelope
for Target System

Black Knight and ARV
Autonomous Vehicle Test Bed

- Refresh perception and navigation modules using sensor advancements.
- Introducing autonomy into the system hierarchy introduces complexity:
  - Vehicle, Tactical and Operational perception and decision making: leveraging advances in learning, behavior composition, formal methods.
- Need an agile lifecycle process: concept – design – build – deploy.

Autonomous System V&V
Notional Approach

- Validate base system (no learning has occurred).
- Verify that learning algorithms can only update interior behavior of well-defined components.
  - Learning will not rewrite the overall control logic of the system as a whole.
- This preserves the structure of the validation argument for the base system.
  - Learning cannot "rewire" the system to either add or remove I/O channels.
- Safety envelope is preserved by runtime behavior verification provided by the: Intent Safety Guard.
  - Intent specification is a broad description of overall required system behavior.
  - Underspecifies behavior significantly, admitting a wide range of possible implementations – pre-coded and learned.



Army Maneuver Center of Excellence (MCOE)
Major Alan Stephens

THE SEVENTEENTH ANNUAL
HIGH CONFIDENCE SOFTWARE AND SYSTEMS CONFERENCE
Annapolis, MD | May 8-10, 2017