An Environment for Comparing Command and Control Architectures

"Know the enemy, know yourself; your victory will never be endangered. Know the ground, know the weather; your victory will then be total"*

"We need improved information assurance standards to enable joint interoperability"**

Dr. John James, Maj Fernando Maymi

John.James, Fernando.Maymi@usma.edu

* **The Art of War** by Sun Tzu , Translated by Samuel B. Griffith, Page 129

** GEN Paul Kern, CG, AMC, plenary speaker, IEEE Information Assurance Workshop, West Point, NY, 18 June 2003

Contents

- Hybrid systems (mixed-signal systems)
  - Command and Control (high-level control)
  - Target Engagement (low-level control)
  - Information Assurance (an enterprise service)
- An Environment for Comparing Command and Control Architectures
  - The OneSAF Objective System (goal: mission rehearsal)
  - The Future Combat System (manned and autonomous systems)
  - The Loitering Attack Missile (autonomous indirect fire)
- Alternative Architectures
  - Centralized control
  - Semi-autonomous control (man-in-the-loop)
  - Autonomous control (missiles select and engages targets)

Command and Control is a Hybrid System e.g. Maintaining Running Estimates

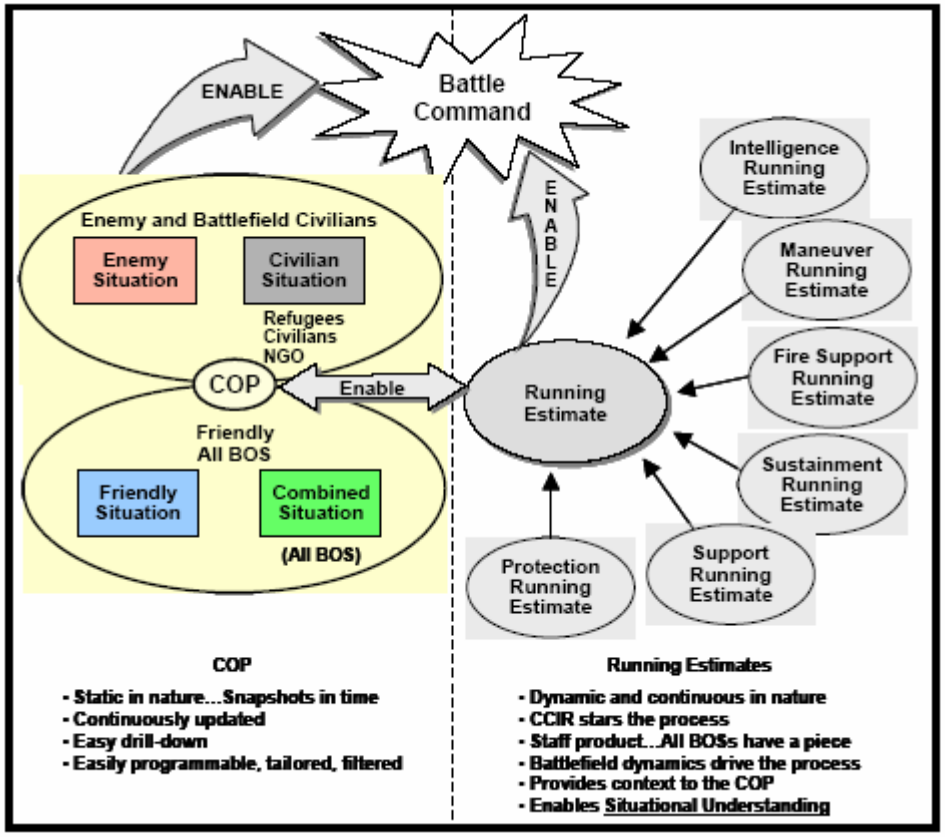


Figure 5-1 of FM 2-0 Intelligence "Common Operational Picture and Running Estimates"

Command and Control is a Hybrid System e.g. Autonomous Missile Control

Common Jam Resistant Digital Targeting

•GPS for accurate search and target location.
•Data link provides targeting coordinates and BDA to operator and allows in-flight missile retasking

Flexible Lethality

Multimode warhead for light armor and soft targets

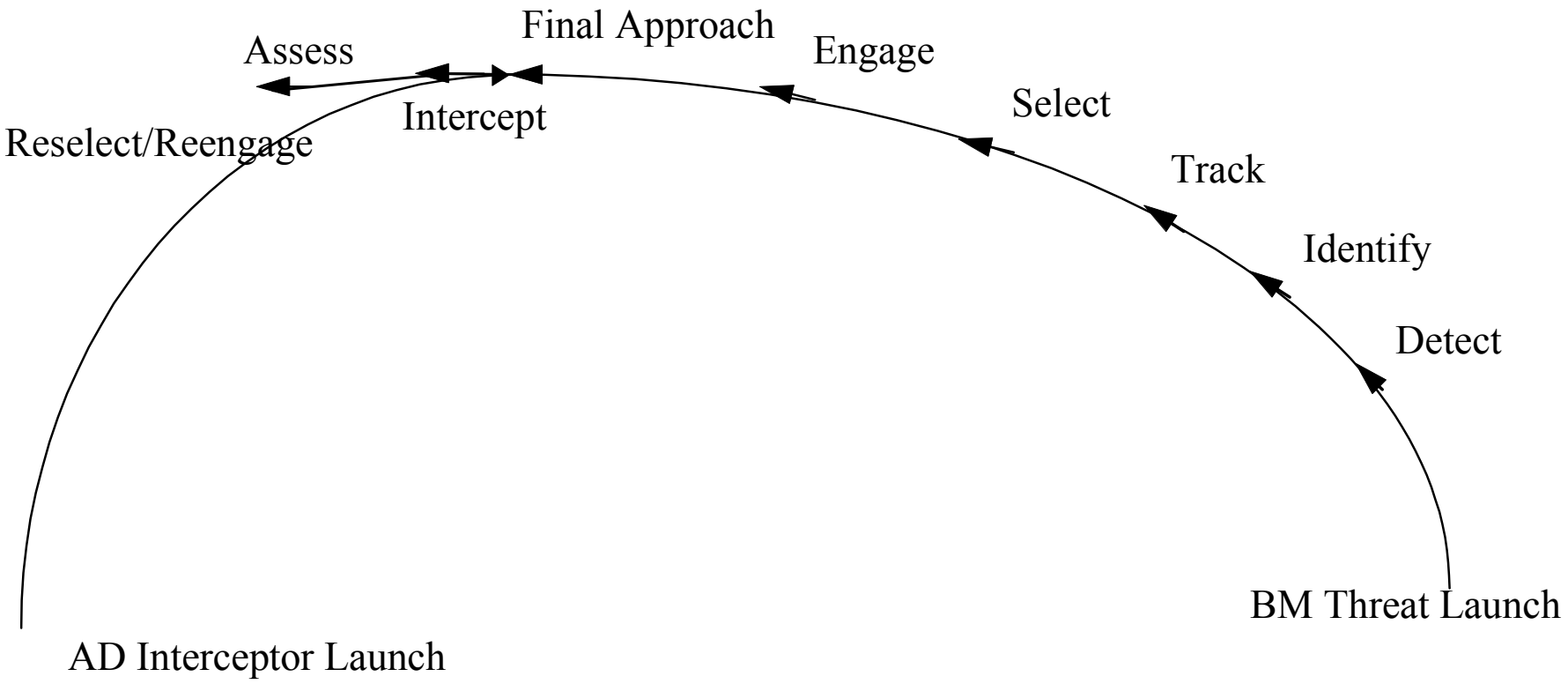
Common Vertical Launch Compatibility

Booster rocket and mini-turbo jet sustainer motor allows launch from same C/LU as PAM

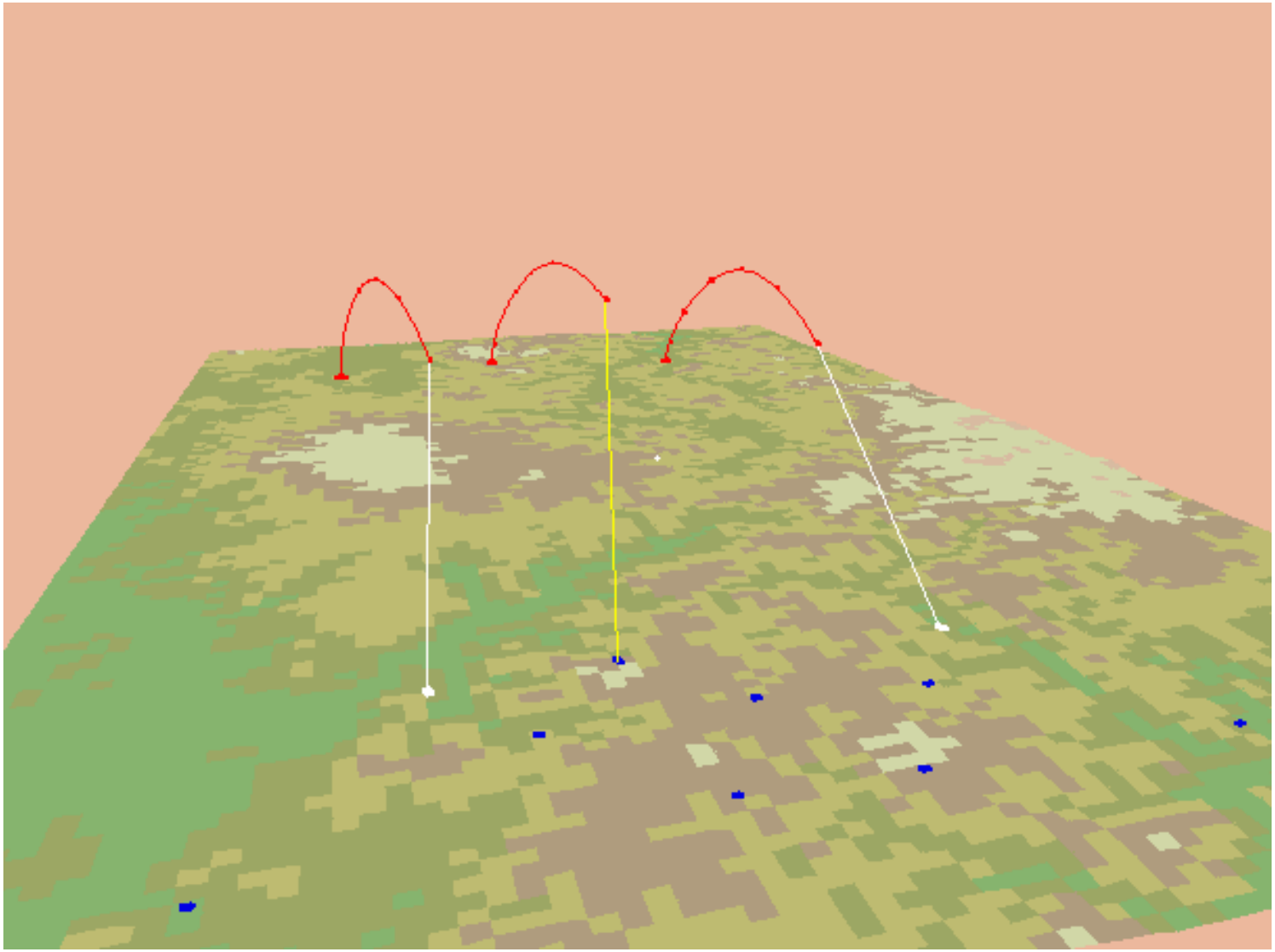
High Capability Seeker

LADAR seeker uses ATR

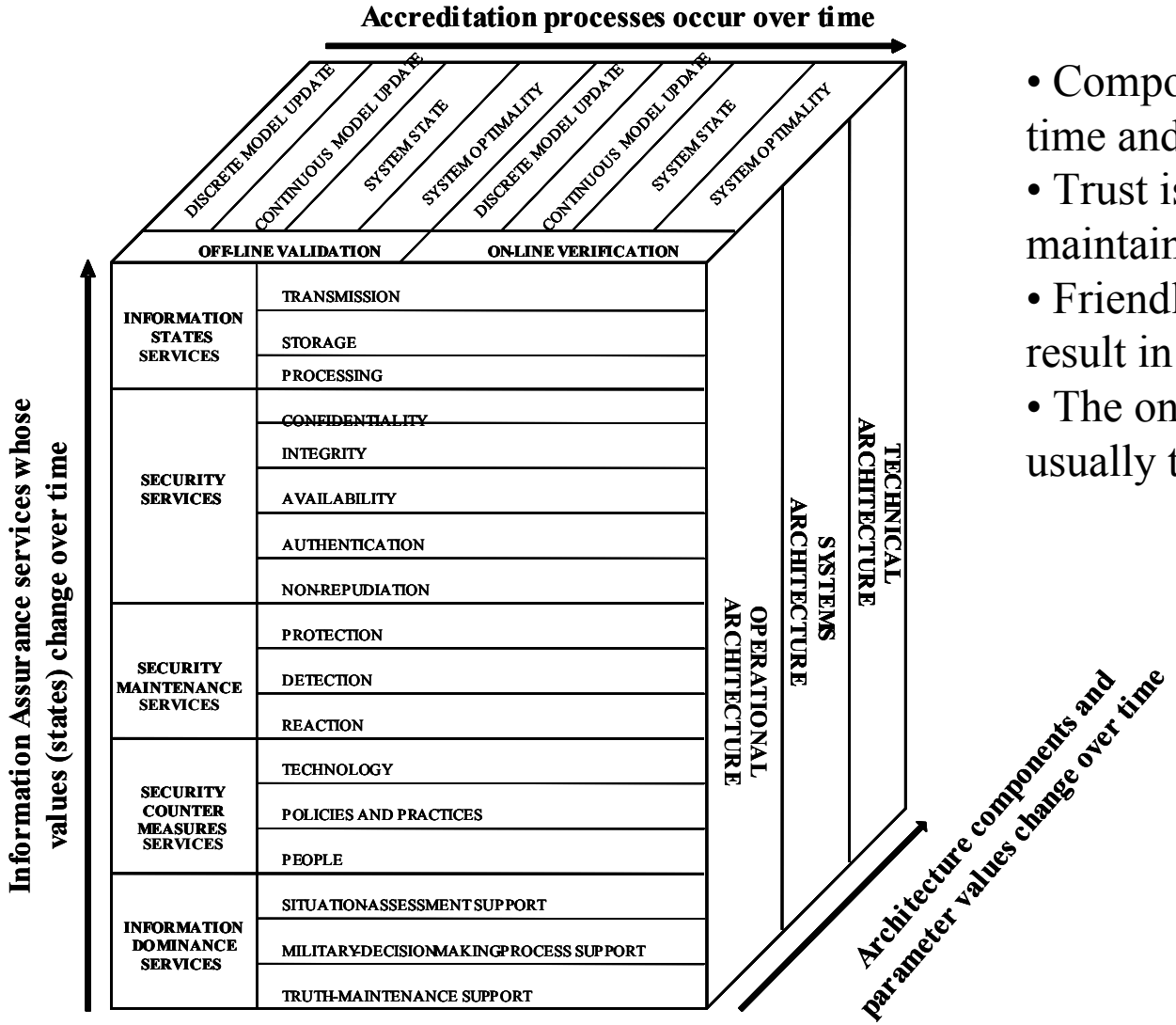
Target Engagements are Hybrid Systems
e.g. Ballistic Missile Engagement Sequence



Ballistic Missile Engagements



Enterprise Services are Hybrid Systems
e.g. Maintenance of trust in distributed components



- Components are distributed in time and space
- Trust is necessarily a locally maintained estimate
- Friendly and enemy activities result in continual system changes
- The only system invariant is usually the commander's intent

An Environment for Comparing Command and Control Architectures

- The OneSAF Objective System (OOS)
  - Will support creation of mission scenarios for various Battlefield Operating Systems (BOS)
  - A goal of OneSAF is to enable mission rehearsal enroute to an operation
- The Future Combat System
  - Will have enhanced connectivity to a variety of information systems
  - Will interface with both manned and autonomous systems
- The Loitering Attack Missile
  - Is the long-duration component of the Non-Line-of-Sight Launch System (NLOS-LS)
  - Will have an autonomous mode of delivering indirect fires
- We will use an eXtensible Markup Language (XML)-based messaging interface to link the different models

Software Architecture Analysis Method

- Characterize a canonical functional partitioning of the domain
- Map the functional partitioning onto the architecture's structural decomposition
- Choose a set of quality attributes with which to assess the architecture
- Choose a set of concrete tasks that test the desired quality attributes
- Evaluate the degree to which each architecture provides support for each task

Modifying SAAM for Real-Time, Distributed Architectures

- Need explicit consideration of communication links between distributed components
- Need explicit consideration of characteristics of feedback loops:
  - Need to analyze a set of components to determine next state of the set of components
  - Need to let the system "settle-out" before the next perturbation

Architecture Analysis Methodology

- Build a functionally-oriented set of architectures by:
  - Partitioning higher level problem space into task-oriented modules using event-based models
  - Partitioning lower-level problem space into mixtures of event-based modules and Differential Algebraic Equation (DAE) based models
  - Assigning modules to a computational structure (usually a "pipe-and-filter" structure)
  - Establishing communication between modules
- Choose a set of quality attributes with which to assess the architectures (pick success criteria)
- Choose a set of concrete tasks which test the desired quality attributes, and
- Evaluate the degree to which each architecture provides support for each task

A Missile Messaging Structure

```
<?xml version="1.0" ?>
  <!DOCTYPE messages (View Source for full doctype...)>
 <messages>
       <!-- Message from missile network to simulation environment.
             Missile m1: violate physical laws and reposition as
       indicated
             Missile m2: set a new temporary waypoint to avoid a
       collision
             Missile m3: set a new (permanent) loiter pattern  --
       >
<message command="PUT" messageId="100">
<missile command="godHand" missileId="m1">
  <location lat="22311" lon="56478" alt="350" />
  <velocity north="112" east="0" down="0" />
  </missile>
<missile command="setWaypoint" missileId="m2">
  <waypoint waypointId="0" lat="12345" lon="12345" alt="250" />
  </missile>
<missile command="setWaypoint" missileId="m3">
  <waypoint waypointId="1" lat="13345" lon="13345" alt="250" />
  <waypoint waypointId="2" lat="14345" lon="13345" alt="250" />
  <waypoint waypointId="3" lat="14345" lon="14345" alt="250" />
  <waypoint waypointId="4" lat="13345" lon="14345" alt="250" />
  </missile>
  </message>
       <!-- Message from missile network to simulation
       environment.
       Missile m3: launch  -->
<message command="PUT" messageId="101">
  <missile command="launch" missileId="m3" />
  </message>
  </messages>
```

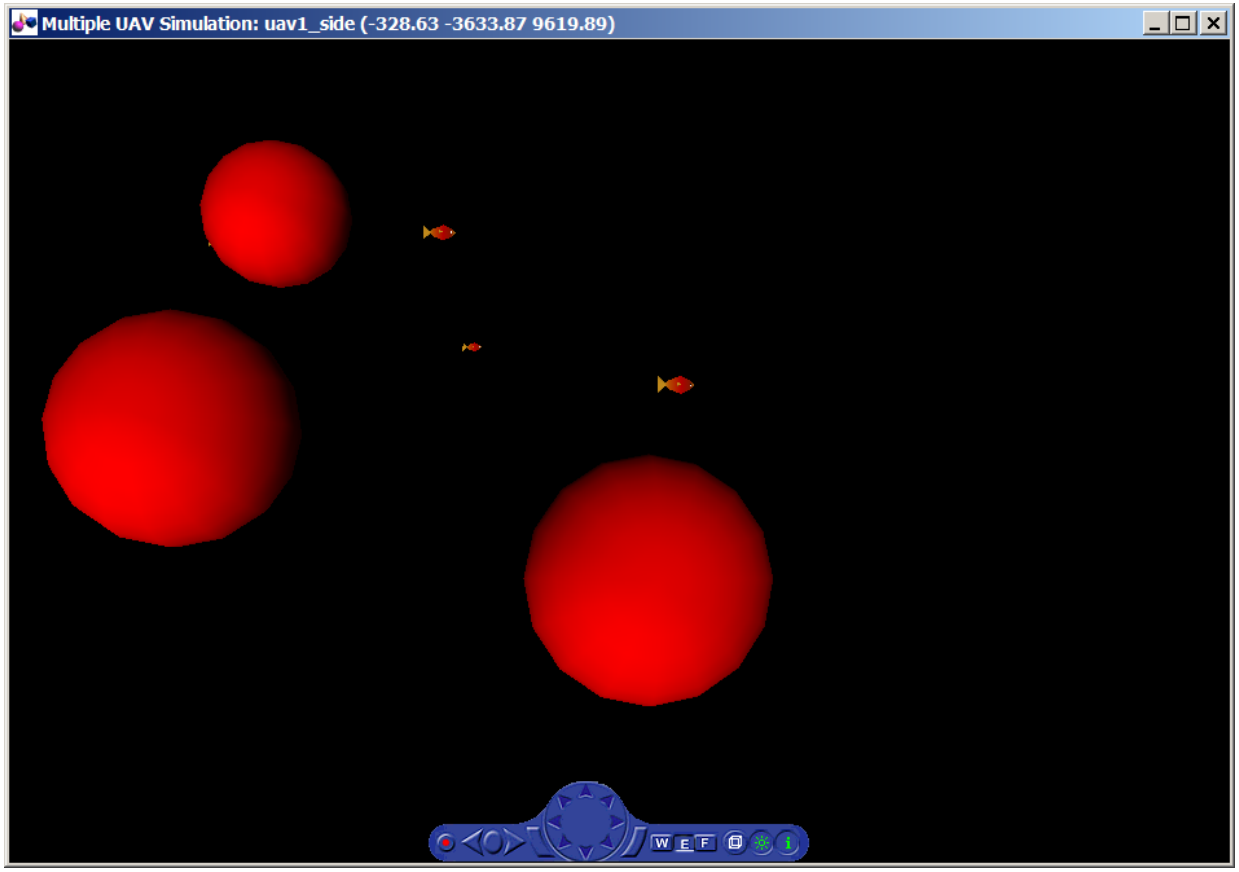
A Missile Messaging Structure (cont.)

```
<?xml version="1.0" ?>
  <!DOCTYPE messages (View Source for full doctype...)>
 <messages>
      <!-- Message from missile network to simulation environment.
            Missile m1: violate physical laws and reposition as indicated
            Missile m2: set a new temporary waypoint to avoid a collision
            Missile m3: set a new (permanent) loiter pattern  -->
      <!-- Message from missile network to simulation environment.

      <!-- Message from missile network to simulation environment.
      Get all update messages for all missiles.  -->
  <message command="GET" messageId="101" />
      <!-- Message from missile network to simulation environment.

      Get all update messages for missile m2.   -->
<message command="GET" messageId="102">
  <missile missileId="m2" />
  </message>
      <!-- Message from missile network to simulation environment.
      Abort (detonate) missile m2.  -->
<message command="PUT" messageId="103">
  <missile command="abort" missileId="m2" />
  </message>
      <!-- Message from simulation environment to missile network.
      Update message for missile m3.           -->
-<message messageId="644" command="get">
-<missile missileId="abc">
  <location lat="13345" lon="13345" alt="250" />
  <velocity north="112" east="0" down="0" />
  </missile>
  </message>
  </messages>
```

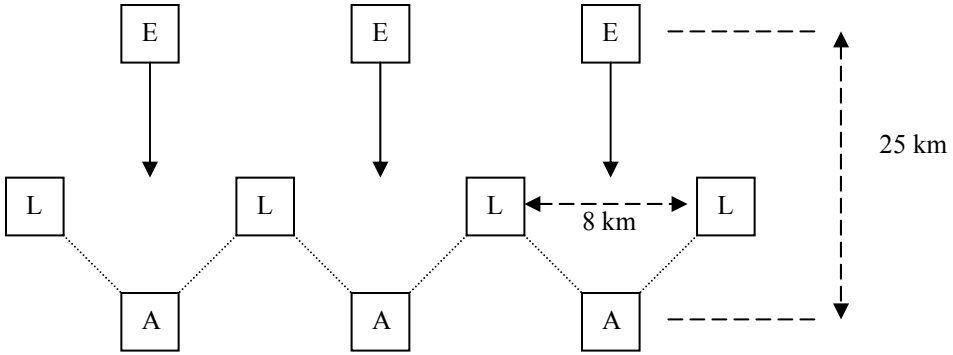
Modeling Swarms of Autonomous Missiles



Matlab-Simulink model of a formation of Multiple UAVs avoiding multiple obstacles to engage targets*

* http://www.mathworks.com/matlabcentral/fileexchange/loadFile.do?objectId=2212&objectType=FILE

Autonomous LAMs



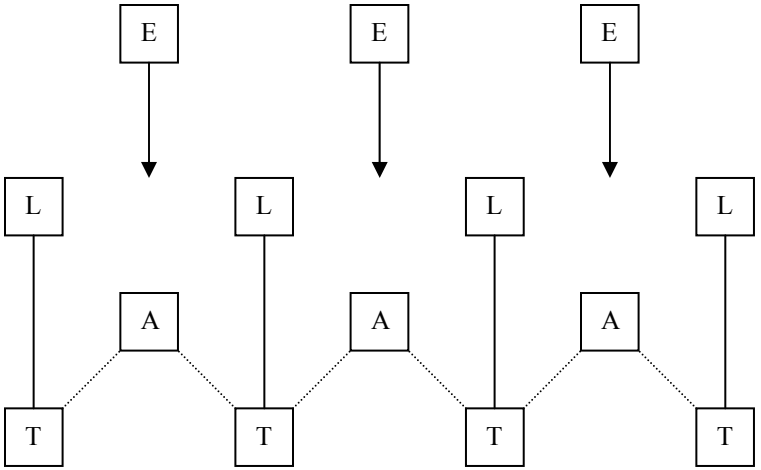
E - Enemy Target (s)

A - Friendly Asset

L - LAM Launcher / Controller

Symbol Key

Autonomous TOCs



E - Enemy Target (s)

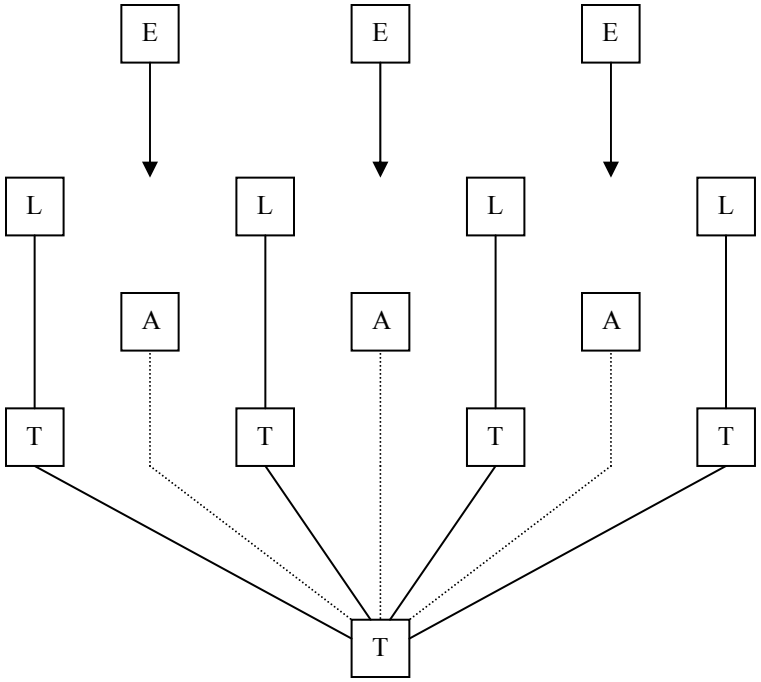
A - Friendly Point Asset

L - LAM Launcher / Controller

T - Tactical Operations Center (TOC)

Symbol Key

Two-Tier Centralized



E - Enemy Target (s)

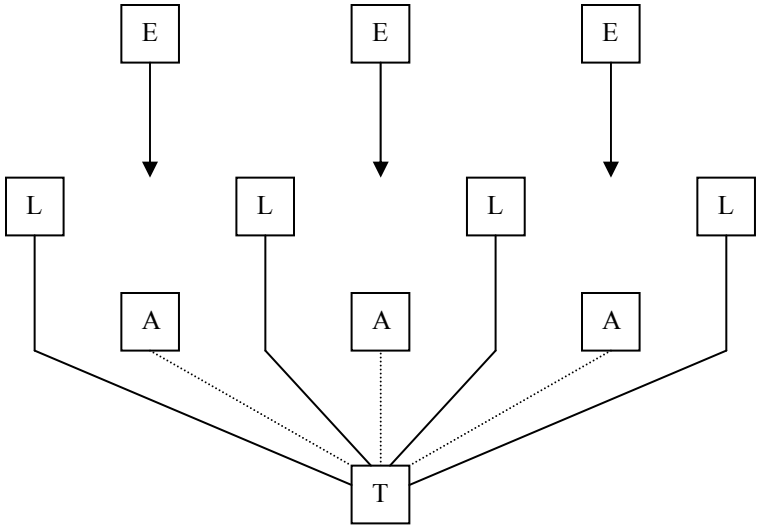
A - Friendly Point Asset

L - LAM Launcher / Controller

T - Tactical Operations Center (TOC)

Symbol Key

One-Tier Centralized



L - Enemy Target (s)

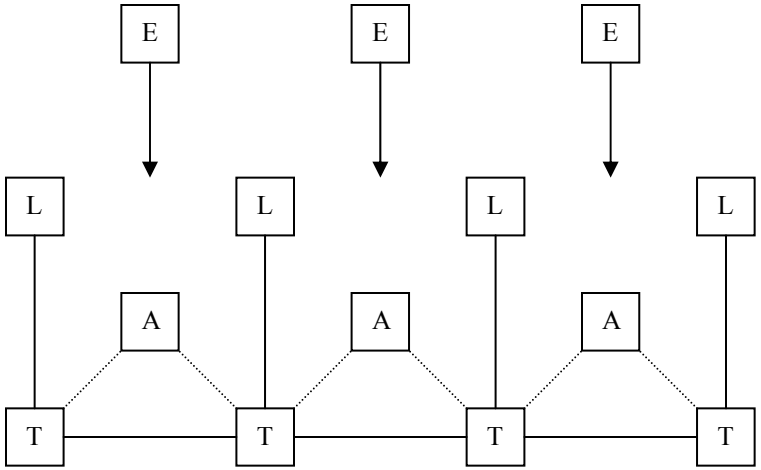
A - Friendly Point Asset

L - LAM Launcher / Controller

T - Tactical Operations Center (TOC)

Symbol Key

Peer-to-Peer Netted



E - Enemy Target(s)

A - Friendly Point Asset

L - LAM Launcher / Controller

T - Tactical Operations Center (TOC)

Symbol Key

Messaging Interface to OneSAF

- The initial version of the Mercury messaging interface to the OneSAF Objective System (OOS) has been completed
  - Interface to cadet code for airspace de-confliction is operational
  - Will be interfaced to the OOS this Summer
- The Matlab Simulink code generates a Virtual Reality Modeling Language (VRML) simulation of three behaviors:
  - Formation keeping,
  - Target seeking, and
  - Collision Avoidance
- The expectation is that a cadet team next year will build on the results of this year to:
  - Evaluate alternative command and control architectures, and
  - Investigate information assurance aspects of the distributed control problem

Summary

- Critical infrastructure processes (such as military operations) are hybrid systems (i.e. have discrete and continuous components)
- Understanding (controlling) complex dynamical processes requires explicit modeling of both discrete and continuous components
- Integration of service and coalition command and control systems will necessarily require on-line, adaptive certification of the level of trust of new system components
- Future command and control systems will feature centralized, semi-autonomous and autonomous control of various joint force components in order to meet enterprise process goals (commander's intent(s))