Discrete Event Component Architecture for Modeling Ships

Arnie Buss
Research Associate Professor
MOVES Institute
abuss@nps.edu

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
JUL 2010		**00-00-2010 to 00-00-2010**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
Discrete Event Component Architecture for Modeling Ships	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Naval Postgraduate School,Moves Institute,Monterey,CA,93943**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES
Research & Education Summit, 13-15 July 2010, Monterey, CA

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**Same as Report (SAR)**	**31**	

Overview

- Objectives
- Sprint Through Component-Based Discrete Event Methodology
- Component Architecture for DES Modeling of Ships
- Status
- Next Steps

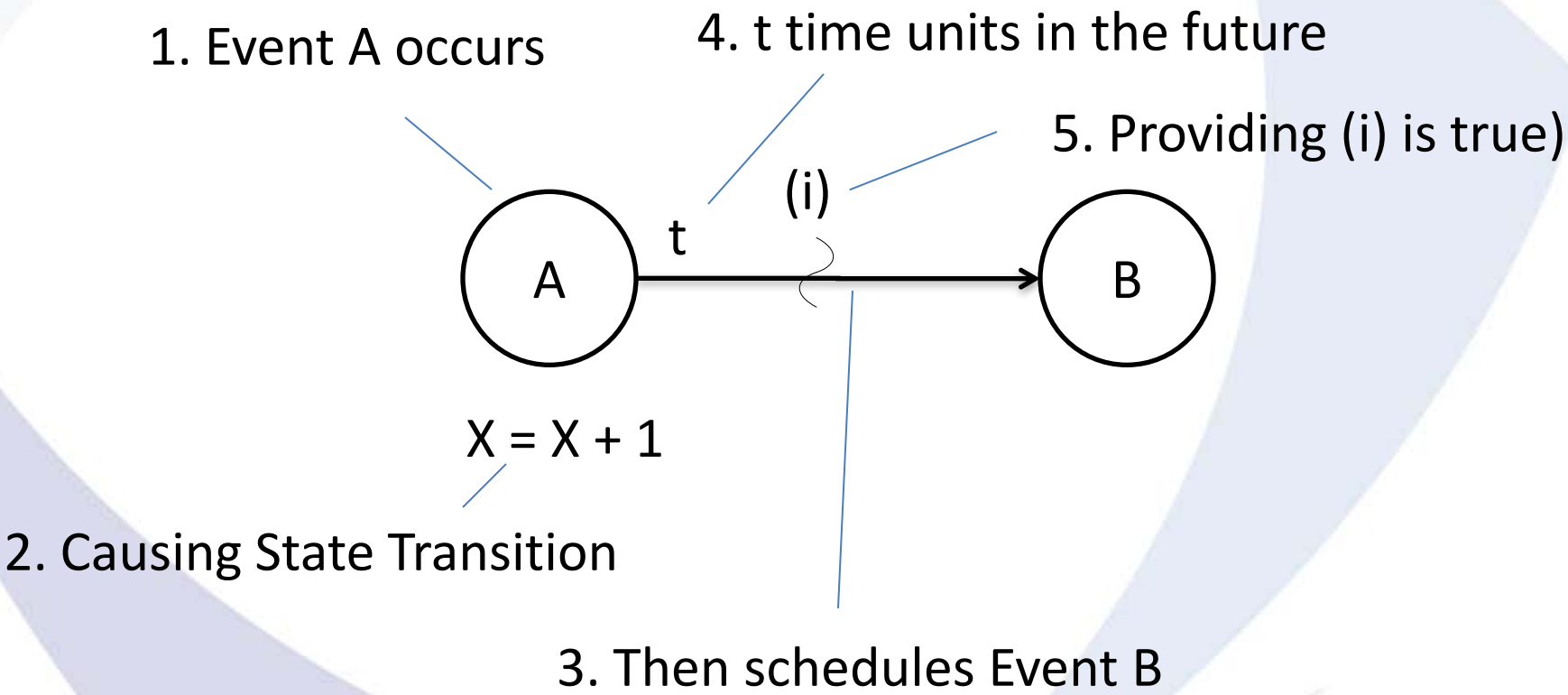
Objectives

- Analysis-oriented
  - Evaluate good or ideal force levels
  - Compare Tactics
  - Evaluate new platforms in operational setting
  - Evaluate new ship systems in operational setting
- Make changes or modifications to existing systems

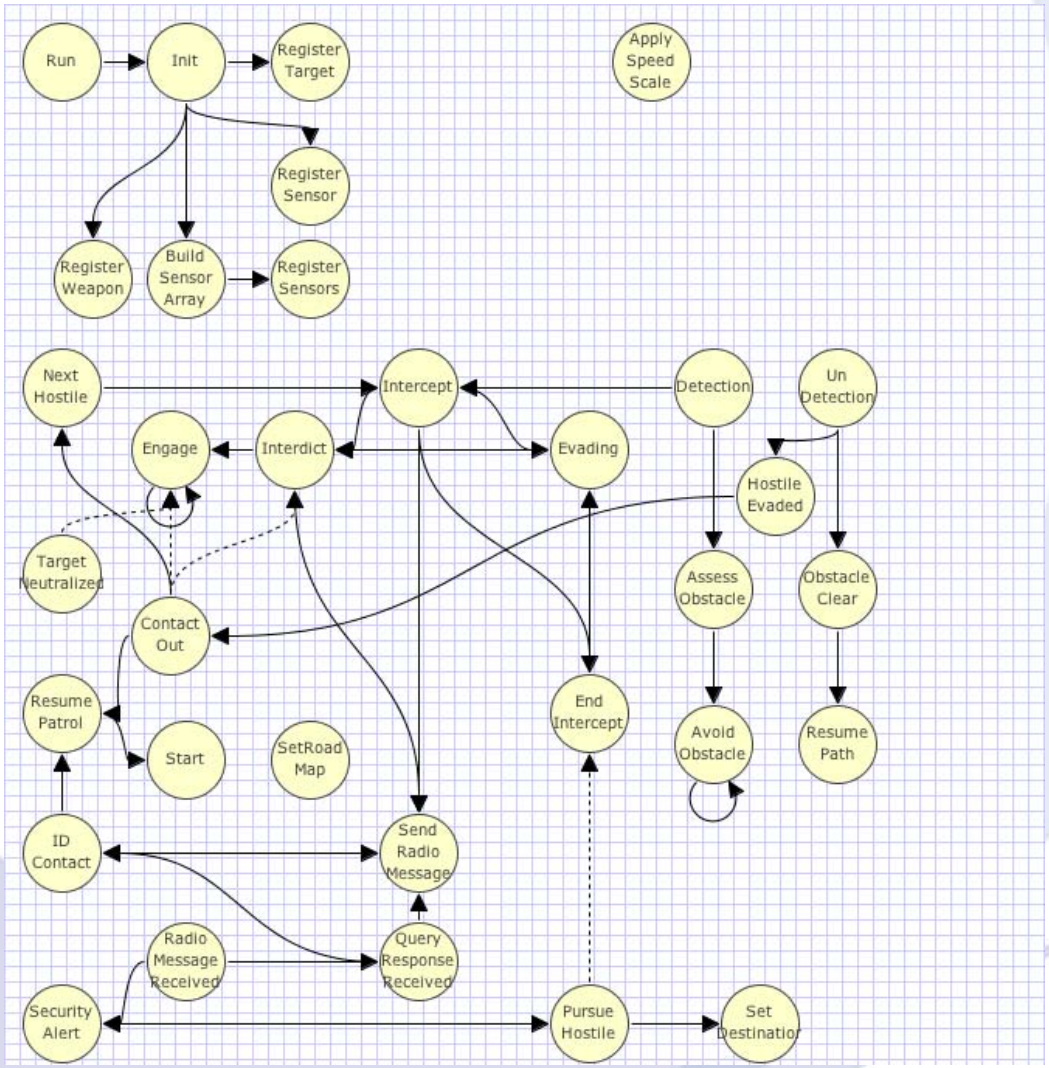
- 

g
tting
- g

Event Graph


1. Event A occurs
4. t time units in the future
5. Providing (i) is true)
(i)
t
A
B
X = X + 1
2. Causing State Transition
3. Then schedules Event B

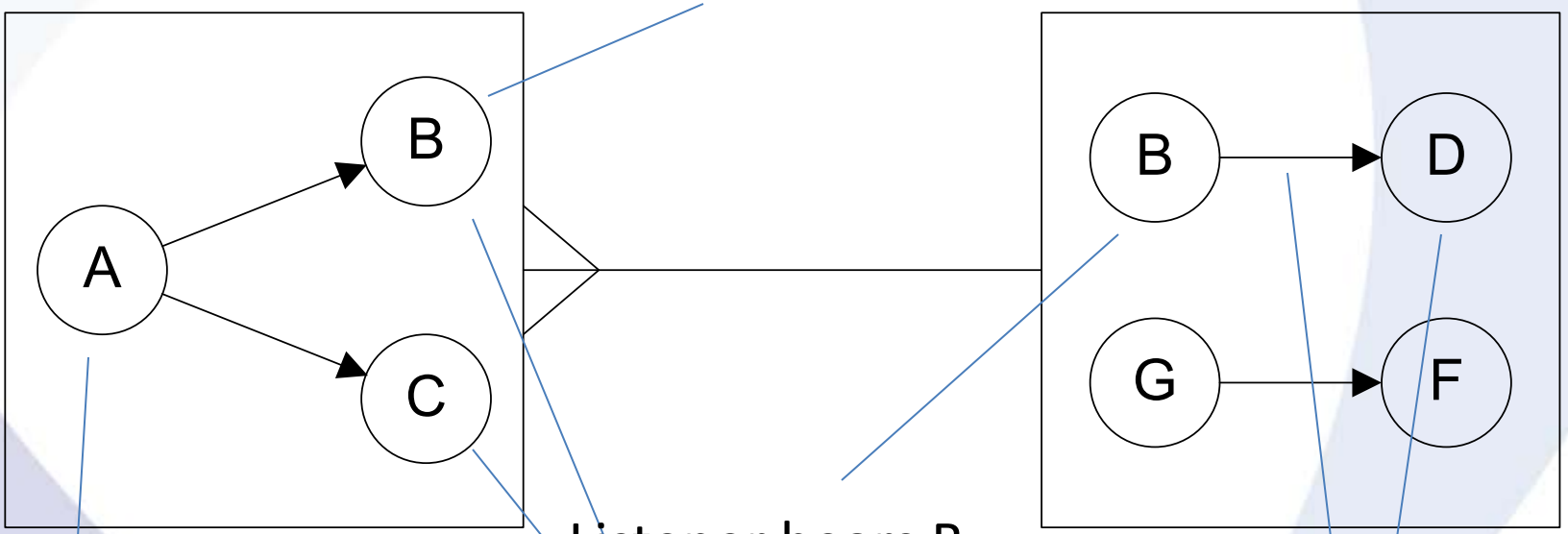
Example


Run
Init
Register Target
Apply Speed Scale
Register Sensor
Register Weapon
Build Sensor Array
Register Sensors
Next Hostile
Intercept
Detection
Un Detection
Engage
Interdict
Evading
Hostile Evaded
Target Neutralized
Assess Obstacle
Obstacle Clear
Contact Out
Resume Patrol
End Intercept
Start
SetRoad Map
Avoid Obstacle
Resume Path
ID Contact
Send Radio Message
Radio Message Received
Query Response Received
Security Alert
Pursue Hostile
Set Destination

Event Graph Components

- Each component encapsulates its own
  - States
  - Event Graph Parameters
  - Event Graph

SimEvent Listening



Advantages of Event Graph Components

- Increases scaleability
- Functional decomposition
- Loose coupling
- Reuse
- Flexibility

Ship Operational Components

- Movement
- Tactics/Behavior
- Sensing
- Weapons
- Communication
- Containers

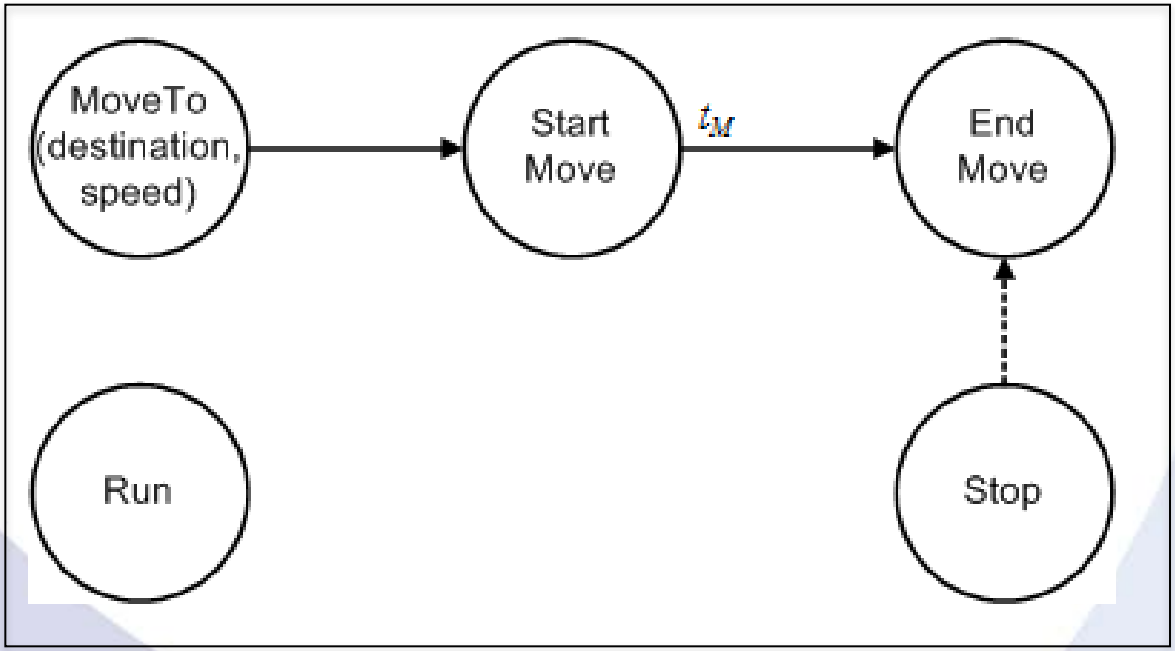
Key Criteria for Level of Detail

- Relevance to questions being asked of model
  - Is it necessary to answer questions?
- Does it impact any estimated measures?
  - Does more detail answer question any better?
- Ideal level of detail
  - As simple as possible
  - But no simpler
- Components allow for simple implementation of multi-level resolution

Modeling Movement

- Location *cannot* be DES state
- However, all movement can be described by an equation of motion
- Example: constant velocity $x(t) = x_0 + (t - t_0)v$
- DES state is initial conditions: (t_0, x_0, v)

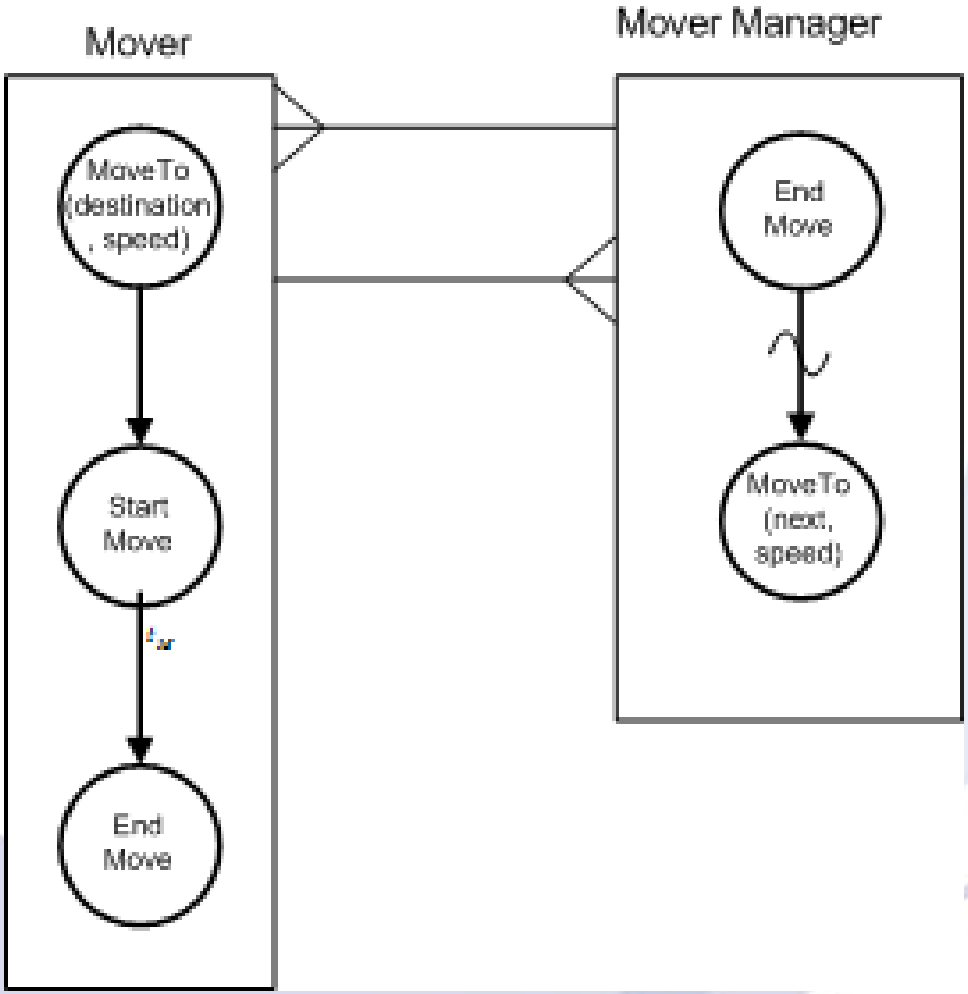
Mover Event Graph Component



Mover Manager

- Separate movement rule from movement logic
- Use listening to schedule next move
- Easy to define new movements rules
- Examples
  - PathMoverManager
  - PatrolMoverManager
  - RandomMoverManager

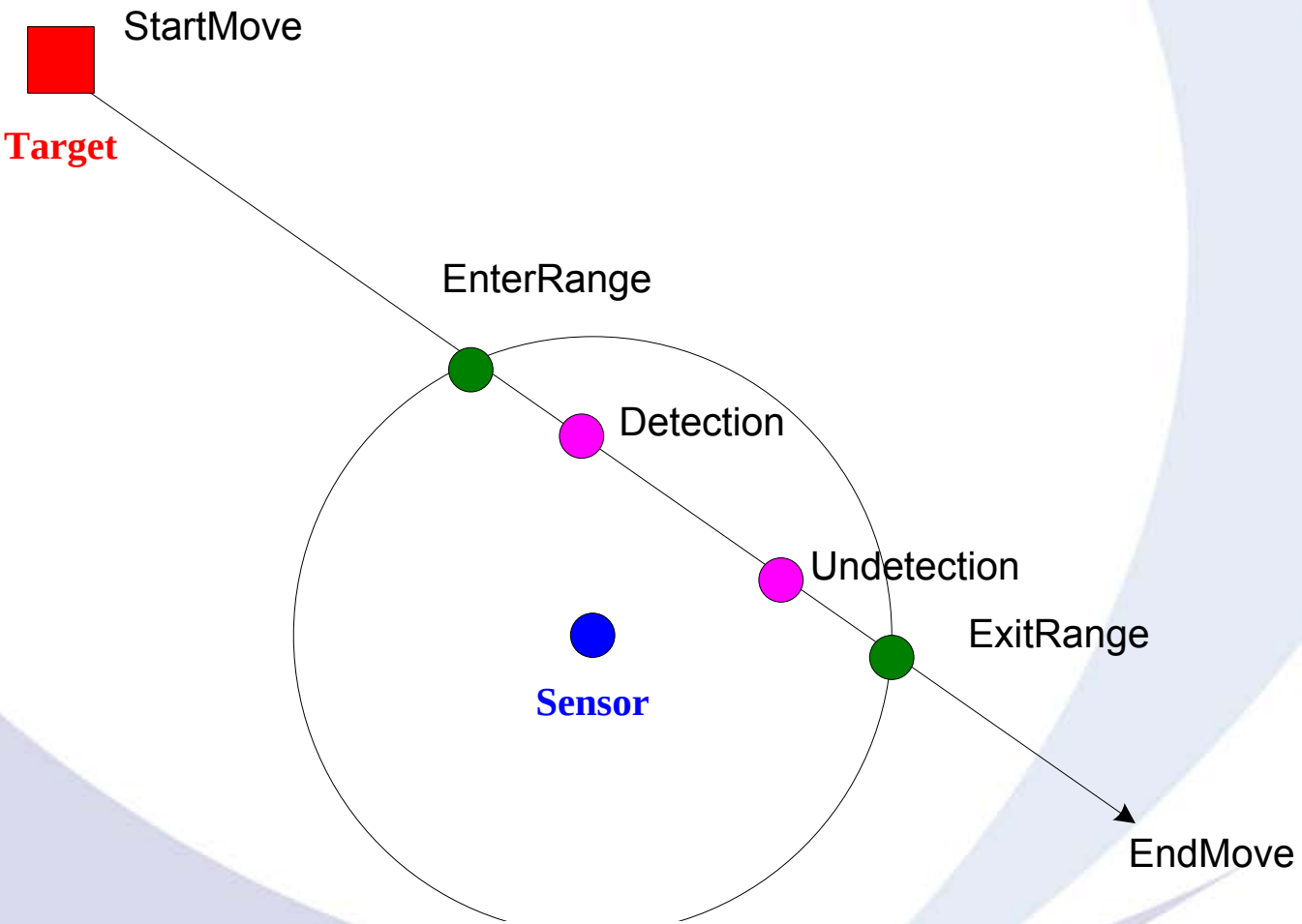
Mover and Mover Manager



Sensing

- Detection only possible within “maximum range” of sensor
- Outside range, no interactions
- Canonical Event Sequence:
  - Enter Range
  - Detection
  - Undetection
  - Exit Range

Canonical Sequence



For Uniform Linear Motion

- Time to Enter/Exit Range:

$$t = -\frac{x \cdot v}{\|v\|^2} \pm \frac{\sqrt{\|v\|^2 (R^2 - \|x\|^2) + (x \cdot v)^2}}{\|v\|^2}$$

Detection

- After Range is entered, Detection occurs sometime later
- Examples
  - Cookie Cutter: delay = 0.0
  - Constant Rate: delay ~ Exp(λ)
  - Other distributions (e.g. Gamma)
  - Glimpse: Every Δt, Detection w/prob p(…)

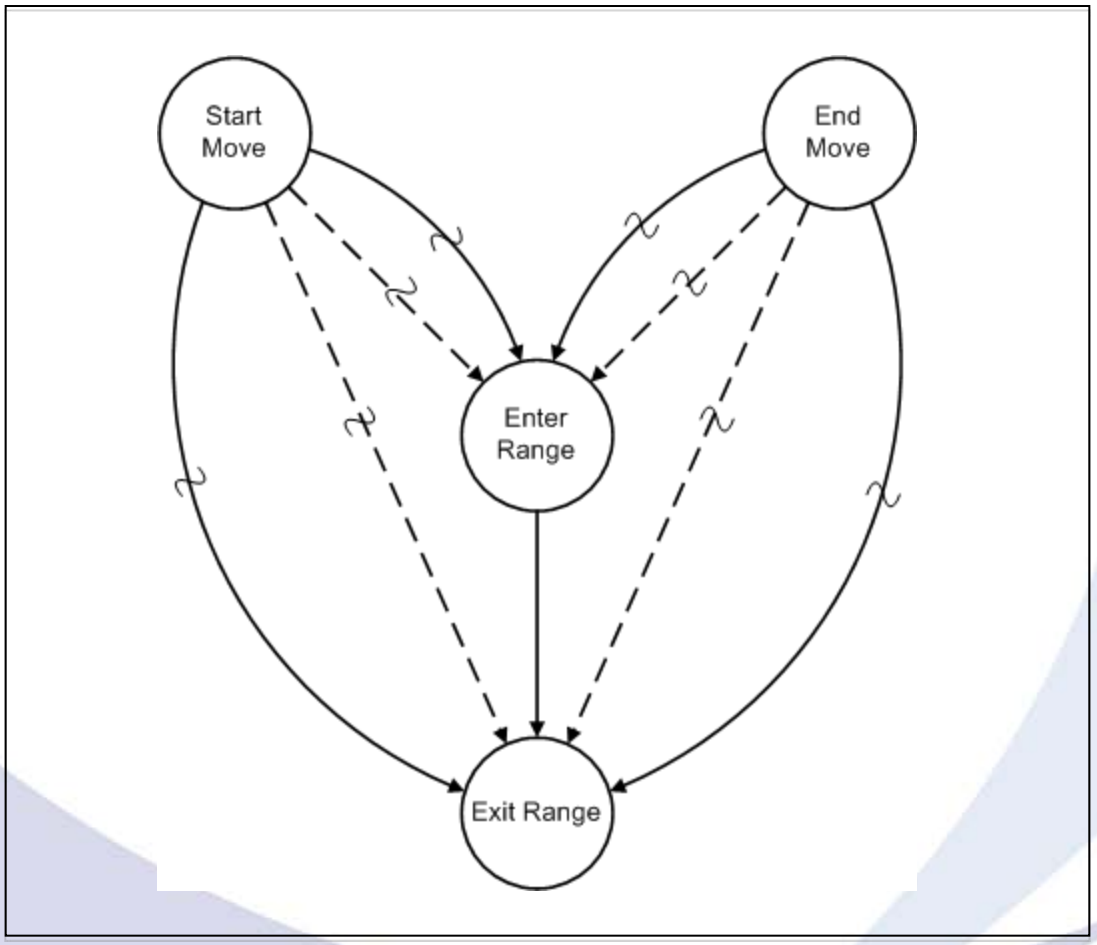
Sensing Framework

- Three types of objects
  - Sensor classes
  - Referee class
  - Mediator classes
- Referee responsible for Enter/Exit Range
- Mediator responsible for Detection/Undetection
- Each Sensor/Target/Mediator triple implements a detection algorithm

Referee

- Responsible for computing & scheduling EnterRange and ExitRange events for all registered sensor/target pairs
- EnterRange events cause the appropriate Mediator to be tasked with adjudicating the actual detection
- Multiple instances of Referee can capture different “bandwidths”

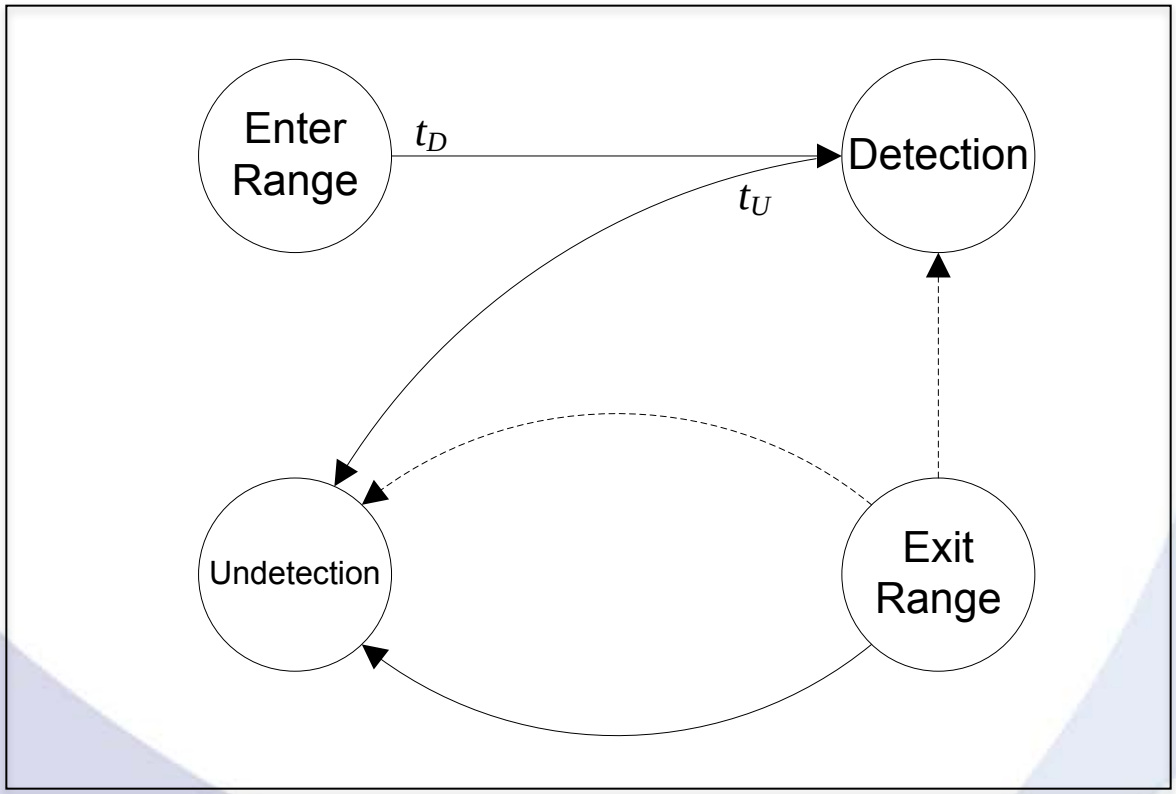
Referee Event Graph



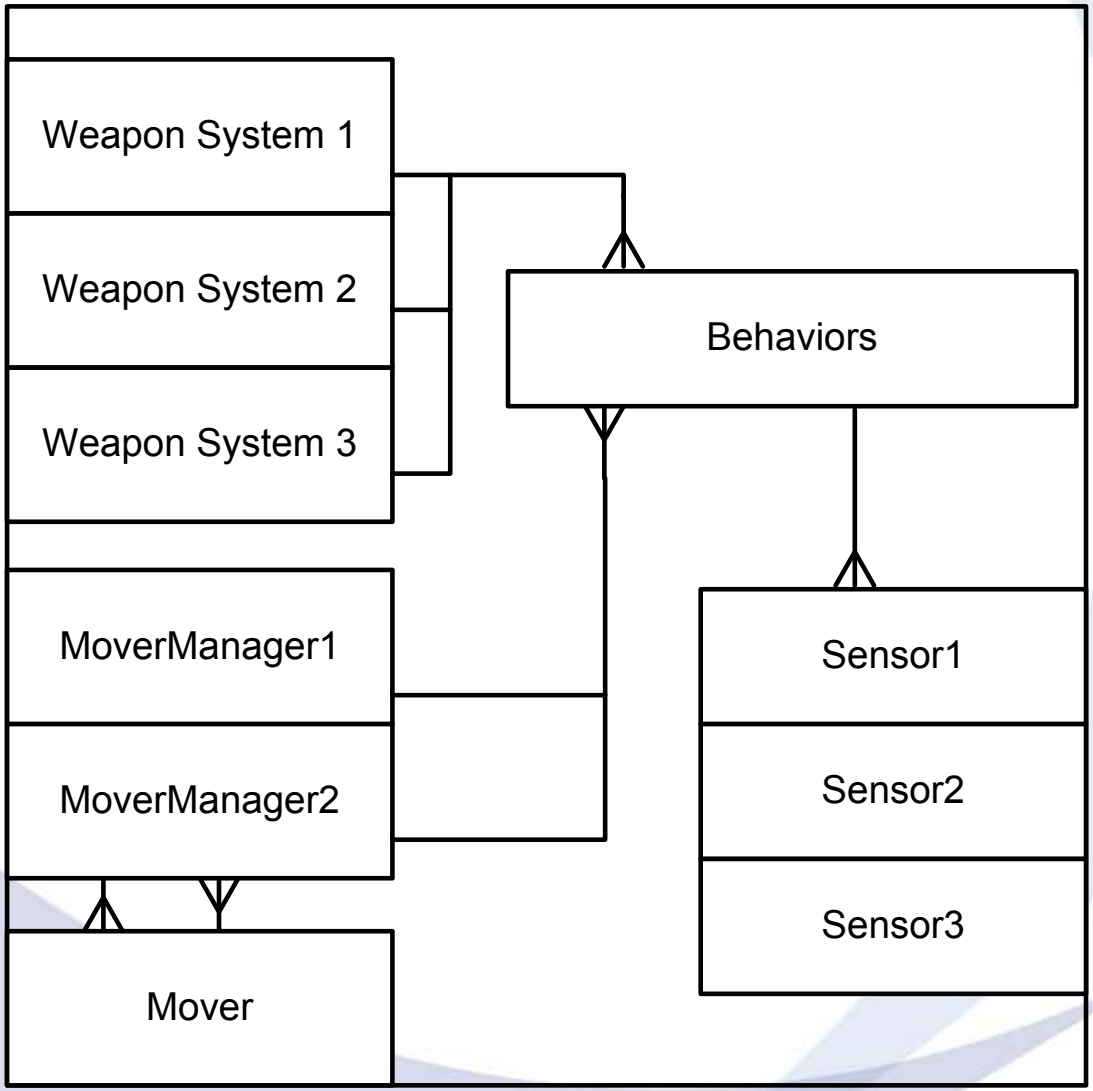
Mediator

- Can implement different detection algorithms for every Sensor/Target pair
  - Can have given Sensor use one algorithm for one type of target and another for a different type of target
- Simple to implement
- Configure MediatorFactory
- Makes implementing new algorithms easy

Mediator Event Graph



Basic Organization of Platform



Multiple Weapons Systems

Behaviors based on perception

Multiple Mover Managers. (Only 1 Active at any time.)

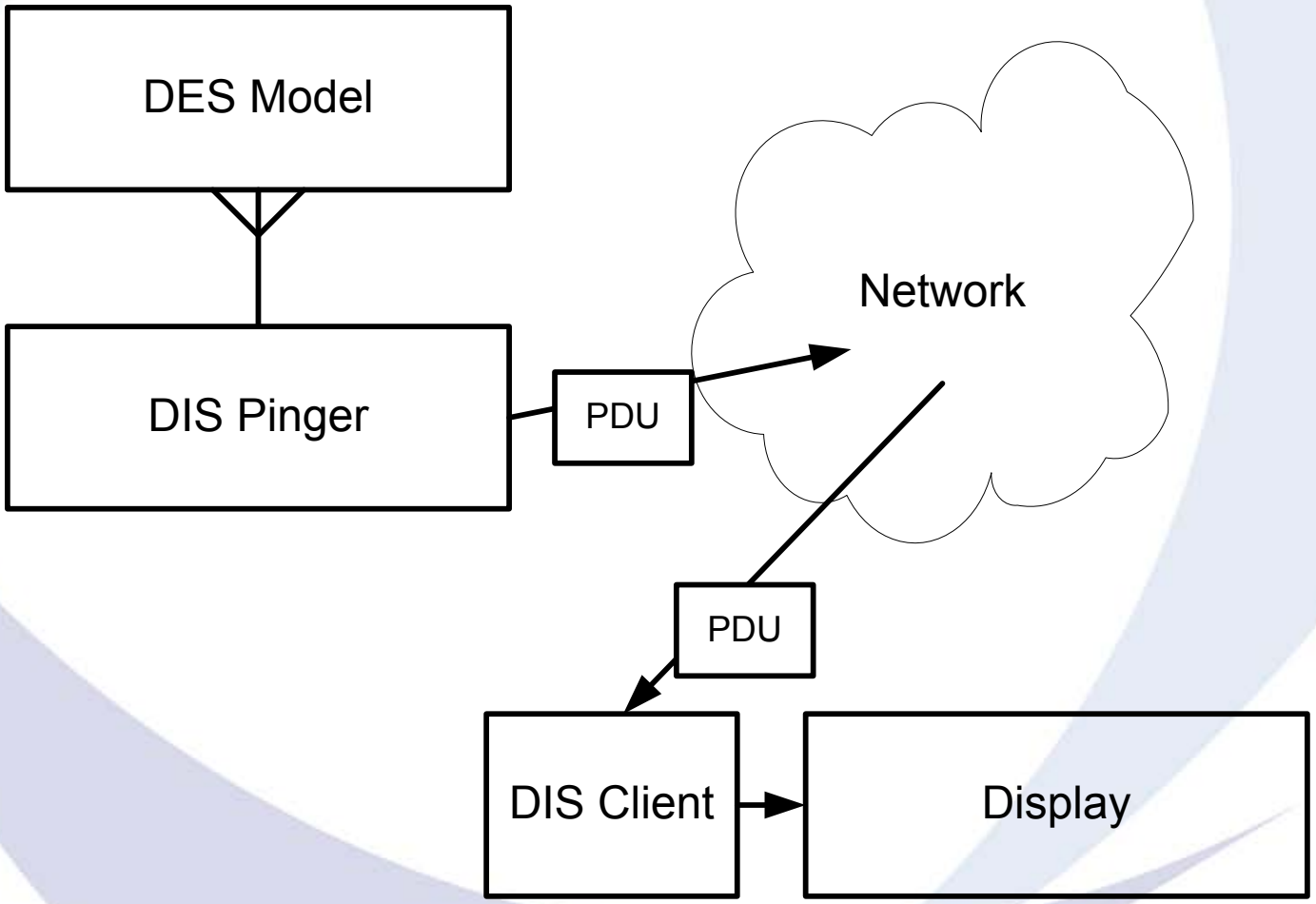
Multiple Sensors

One Mover Component

Implementation

- Simkit (Java)
- Viskit (XML/Java)
- SimPykit (Python)

Loose Coupling Between Model and Display (DIS)



Client Displays

- Hand-crafted 2-D (one-offs)
- XJ3D
- Delta3D (Under Construction)
  - Using DIS
  - SimPykit using Python bindings

Status

- Framework complete for:
  - Movement
  - Movement tactics
  - Sensing
  - Hooks for behavioral response
  - Simple behaviors
- To be done:
  - Complete design of behavioral components
  - Complete first-level implementation (code)

Next Steps

- Continue work on modeling behavior and tactics
- Complete design of complex components
- First version of TCraft platform as exemplar
- Work towards complete (modeler-friendly) simulation

Questions?

Arnie Buss
Research Associate Professor
MOVES Institute
abuss@nps.edu