Discrete Event Component Architecture for Modeling Ships

Arnie Buss
Research Associate Professor
MOVES Institute
abuss@nps.edu
**Discrete Event Component Architecture for Modeling Ships**

**Performer Organization Name(s) and Address(es)\(^\text{1}\)**

Naval Postgraduate School, Moves Institute, Monterey, CA, 93943

**Performing Organization Report Number\(^\text{2}\)**

**Sponsor/Monitoring Organization(s) and Address(es)\(^\text{3}\)**

**DISTRIBUTION/AVAILABILITY STATEMENT\(^\text{12}\)**

Approved for public release; distribution unlimited

**Supplementary Notes\(^\text{13}\)**

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

<table>
<thead>
<tr>
<th>16. SECURITY CLASSIFICATION OF:</th>
<th>17. LIMITATION OF ABSTRACT</th>
<th>18. NUMBER OF PAGES</th>
</tr>
</thead>
<tbody>
<tr>
<td>a. REPORT</td>
<td>Same as Report (SAR)</td>
<td>31</td>
</tr>
<tr>
<td>b. ABSTRACT</td>
<td>unclassified</td>
<td></td>
</tr>
<tr>
<td>c. THIS PAGE</td>
<td>unclassified</td>
<td></td>
</tr>
</tbody>
</table>

\[^{1}1\text{Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.}\]

\[^{2}1\text{REPORT DATE}\]

**JUL 2010**

\[^{3}1\text{REPORT TYPE}\]

**3. DATES COVERED**

**00-00-2010 to 00-00-2010**

\[^{4}1\text{TITLE AND SUBTITLE}\]

**Discrete Event Component Architecture for Modeling Ships**

\[^{5}1\text{AUTHOR(S)}\]

**Naval Postgraduate School, Moves Institute, Monterey, CA, 93943**

\[^{6}1\text{SPONSOR/MONITOR'S ACRONYM(S)}\]

**Research & Education Summit, 13-15 July 2010, Monterey, CA**

\[^{7}1\text{ABSTRACT}\]

**15. SUBJECT TERMS**

\[^{8}1\text{NUMBER OF PAGES}\]

**31**

\[^{9}1\text{RESPONSIBLE PERSON}\]

**Standard Form 298 (Rev. 8-98)**

Prescribed by ANSI Std Z39-18
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
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
Event Graph

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

\[ X = X + 1 \]
Example
Event Graph Components

• Each component encapsulates its own
  – States
  – Event Graph Parameters
  – Event Graph
SimEvent Listening

When Event B occurs

Listener hears B

Event A Occurs

Scheduling B & C

Listener "hears" A
But has no match

And schedules D

87/14/2010 2010 MOVES Research Summit
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
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 $\sim$ Exp($\lambda$)
  - Other distributions (e.g. Gamma)
  - Glimpse: Every $\Delta t$, 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

- Enter Range
- Detection
- Undetection
- Exit Range

$t_D$ and $t_U$ transitions between states.
Basic Organization of Platform

- Multiple Weapons Systems
- Multiple Mover Managers. (Only 1 Active at any time.)
- One Mover Component

Behaviors based on perception

Multiple Sensors
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