Data Distribution

Dynamic Data Distribution

Richard M. Fujimoto
Professor

Computational Science and Engineering Division
College of Computing
Georgia Institute of Technology
Atlanta, GA 30332-0765, USA

http://www.cc.gatech.edu/~fujimoto/

Copyright © Richard M. Fujimoto
Outline

• Introductory Comments
• Dynamic (Value based) Data Distribution: HLA Data Distribution Management
  – Routing space
  – Publication Region
  – Subscription Region
• DDM Implementation
  – Cell-Based
  – Region-Based
  – Combining Cells and Regions
Background

• Basic question: When a simulator generates information (e.g., state updates) that may be of interest to other simulators, who should receive the message?

• Example: moving vehicles in a virtual environment
  – Moving vehicle sends “update” messages indicating new position
  – Each vehicle that can “see” the moving vehicle should receive a message
  – How does the sender/RTI know which other federates should receive the message?
    • Data distribution is essentially a message routing problem
Communication Primitives

- **Unicast**
  - One sender, message received by one destination
- **Broadcast**
  - One sender, message received by all destinations
- **Multicast**
  - One sender, message received by multiple (but not necessarily all) destinations
  - Operations (analogous to newsgroups)
    - Join group
    - Leave group
    - Send message to group
  - Can be implemented by unicast, or network multicast
  - Best effort vs. reliable multicast
Using a Grid to Capture Locality

- Divide play-box into non-overlapping (rectangular, hexagonal) grid cells
- Create one multicast group for each cell
- Subscribe to cell(s) you can “see”
- Send message to cell where the vehicle resides
- Requires additional filtering at the receiver

Sensor S joins groups 8 and 9
Vehicles V1 and V2 send to group 8
Vehicle V3 sends to group 5

multicast group, id=15
HLA Data Distribution Management (DDM)

- HLA DDM provides a more general mechanism
- Name Space
  - Routing space: N-dimensional coordinate system
  - Separate from simulation state, used solely for routing
- Interest expressions
  - Subscription region: N-dimension rectangular in routing space
  - Associate region with subscription requests
- Description expressions
  - Update region: N-dimensional rectangle in routing space
  - Associated with each object instance
- A message updating an attribute of an object instance is routed to a federate if:
  - The federate is subscribed to the object’s class and attribute, and
  - The update region associated with the updated attribute overlaps with the federate’s subscription region for that class/attribute
Name Space

- N dimensional routing space
  - Playbox in virtual environment
  - Radio channels for wireless communication

Interest expressions
- Subscription region in routing space (S1=[0.1,0.5], [0.2,0.5])
  - Specifies portion of routing space of interest to federate

- Federate 1 (sensor): subscribe to S1
- Federate 2 (sensor): subscribe to S2
- Federate 3 (target): update region U

Description expressions
- Update region in routing space (U)
- Associated an update region with each attribute update
- A federate receives a message if
  - It has subscribed to the attribute(s) being updated, and
  - Its subscription region overlaps with the update region

Update messages by target are sent to federate 1, but not to federate 2
Update Regions vs. Points

- Routing space represents playbox
- Subscription region represents sensor
- Updates correspond to position of a moving vehicle

- Update points: Sensor not notified of vehicle
- Update regions: Sensor is notified of vehicle
In general, DDM is a compromise among:

- Filtering accuracy
- Implementation considerations (mapping to multicast groups)
- Ease of use

- Vehicle out of range, but updates are still routed to sensor federate
- Messages must be filtered at the receiver
- Sensor range may not be rectangular
Routing spaces (name space) and regions
- Define routing spaces in federation initialization file
- Create Region, Modify Region, Delete Region
  - Used for both subscription and update regions

Subscription regions (interest expressions)
- Subscribe/Unsubscribe Object Class Attributes with Region
  - Used in addition to class-based filtering

Update regions (description expressions)
- Register Object Instance with Region or Associate Region with Updates
- Unassociate Region for Updates
- Update Attribute Values
Outline

• Introductory Comments
• Dynamic (Value based) Data Distribution: HLA Data Distribution Management
  – Routing space
  – Publication Region
  – Subscription Region
• DDM Implementation
  – Cell-Based
  – Region-Based
  – Combining Cells and Regions
Communication Services

Recall:

• Unicast: point-to-point communication
• Broadcast: send to all
• Multicast: send to multiple destinations (not necessarily all)
  – Multicast group
  – Join group, Leave group
  – Send: transport message to every member of group

• Data distribution software must map name space, interest expressions, and description expressions to group communication services
Implementation Approach

• Map name space to multicast groups
  – Not all points in name space need be mapped to groups
  – A point in name space could map to multiple groups

• Interest expression
  – Interest expression defined as points of name space
  – Join groups that overlap with interest expression

• Description expression
  – Description expression defined as points in name space
  – Send messages to groups that overlap with description expression
Grid-Based Implementation

- partition routing space into grid cells, map each cell to a multicast group
- subscription region: *Join* each group overlapping subscription region
- attribute update: send *Update* to each group overlapping update region
- need additional filtering to avoid unwanted messages, duplicates
### Changing a Subscription Region

- **Leave group**
- **Join group**
- **no operations issued**

#### Operations

- **Leave**: operations for (cells in old region - cells in new region)
- **Join**: operations for (cells in new region - cells in old region)

#### Table

<table>
<thead>
<tr>
<th></th>
<th>33</th>
<th>34</th>
<th>35</th>
<th>36</th>
<th>37</th>
<th>38</th>
<th>39</th>
<th>40</th>
</tr>
</thead>
<tbody>
<tr>
<td>25</td>
<td>26</td>
<td><strong>27</strong></td>
<td>28</td>
<td><strong>29</strong></td>
<td>30</td>
<td>31</td>
<td>32</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>18</td>
<td><strong>19</strong></td>
<td><strong>20</strong></td>
<td><strong>21</strong></td>
<td>22</td>
<td>23</td>
<td>24</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>10</td>
<td><strong>11</strong></td>
<td><strong>12</strong></td>
<td><strong>13</strong></td>
<td>14</td>
<td>15</td>
<td>16</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td><strong>3</strong></td>
<td><strong>4</strong></td>
<td><strong>5</strong></td>
<td><strong>6</strong></td>
<td><strong>7</strong></td>
<td><strong>8</strong></td>
<td></td>
</tr>
</tbody>
</table>
Approach 2: Region-Based Groups

- Define one multicast group per publication region
- Group membership: Any federate subscribed to a region that overlaps the publication region is a member of group
- Update: Send message to group associated with publication region

Group for P1:
- federates subscribed to regions S1, S2

Group for P2:
- federates subscribed to region S3
The Matching Problem

• When a subscription region changes, it must be compared against all publication regions to determine if the federate should join/leave multicast groups.

• When a publication region changes, it must be compared against all subscription regions to determine the new composition of its multicast group.

• Not scalable to large numbers of regions.

Changing P1: must compare P1 against S1, S2, and S3 to determine the new composition of P1’s group.
Approach 3: Regions with Grids

- A group is defined for each publication region (same as region-based approach)
- A grid is superimposed over routing space
- Matching: need only check publication/subscription regions in the grid cell(s) overlapping the original and new regions

Changing P1: must compare P1 against S1 and S2, but need not compare against S3
Practical Problems

• May be a limited number of multicast groups

• Fast movers: rapid joins and leaves
  – Join/leave times may be large
  – Predict and initiate group operations in advance

• Wide area viewers: too much traffic!
  – need less detailed information to reduce traffic
  – Multiple routing spaces with different grid sizes and detail of information covering playbox
Summary

• Data distribution management provides value-based filtering of data
  – Dynamic interest, description expressions
  – Design involves many tradeoffs
    • Filtering efficiency
    • Ease of use
    • Implementation complexity

• Implementation
  – Map name space to multicast groups
  – Map interest expressions to multicast group joins
  – Map declaration expression to multicast group sends
  – Interest expression changes map to group joins and leaves