Distributed Virtual Environments

Introduction

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

• What are they?
• DVEs vs. Analytic Simulations
• DIS
  – Design principles
• Example
Distributed Virtual Environments (DVE)

- A synthetic world into which humans and/or physical devices are embedded
- Interaction between embedded humans/devices and simulated elements
- Involves humans, devices, computations at different locations
- Examples
  - Military training (SIMNET, Distributed Interactive Simulation, HLA)
  - Multiplayer video games
- A key issue is to ensure different participants have consistent views of the DVE
  - Consistent in time and space
  - “Fair fight” issues
  - Latency, limited communication bandwidth
DVE Architectures

Server architecture

Distributed architecture

WAN interconnect

LAN interconnect
## Analytic vs. Training

<table>
<thead>
<tr>
<th></th>
<th>Analytic</th>
<th>DVE</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Communication</strong></td>
<td>Often point-to-point</td>
<td>Broadcast/multicast</td>
</tr>
<tr>
<td></td>
<td>Reliable</td>
<td>Best effort</td>
</tr>
<tr>
<td></td>
<td>Multiprocessor/LAN</td>
<td>LAN/WAN</td>
</tr>
<tr>
<td></td>
<td>OK w/ Arbitrary latency</td>
<td>Latency bounds, low jitter</td>
</tr>
<tr>
<td><strong>Simulation Model</strong></td>
<td>May be non-interactive</td>
<td>Interactive</td>
</tr>
<tr>
<td><strong>Time Management</strong></td>
<td>Time stamp order</td>
<td>Receive order</td>
</tr>
<tr>
<td></td>
<td>Synchronization protocol</td>
<td>No synchronization protocol</td>
</tr>
<tr>
<td><strong>Performance</strong></td>
<td>As-fast-as-possible</td>
<td>Real-time</td>
</tr>
<tr>
<td></td>
<td>Speedup</td>
<td>Realism</td>
</tr>
<tr>
<td><strong>Typical Applications</strong></td>
<td>Design, analysis</td>
<td>Training, entertainment</td>
</tr>
<tr>
<td><strong>Issues (selected)</strong></td>
<td>Efficient execution</td>
<td>Scalable execution</td>
</tr>
<tr>
<td></td>
<td>Ease of use</td>
<td></td>
</tr>
</tbody>
</table>
Distributed Interactive Simulation (DIS)

“The primary mission of DIS is to define an infrastructure for linking simulations of various types at multiple locations to create realistic, complex, virtual ‘worlds’ for the simulation of highly interactive activities” [DIS Vision, 1994].

- Developed in the U.S. Department of Defense, initially for training.
- DVEs widely used in DoD; growing use in other areas (entertainment, emergency planning, air traffic control).
DIS Design Principles

• Autonomy of simulation nodes
  – simulations broadcast events of interest to other simulations; need not determine which others need information
  – receivers determine if information is relevant to it, and model local effects of new information
  – simulations may join or leave exercises in progress

• Transmission of “ground truth” information
  – each simulation transmits absolute truth about state of its objects
  – receiver is responsible for appropriately “degrading” information (e.g., due to environment, sensor characteristics)

• Transmission of state change information only
  – if behavior “stays the same” (e.g., straight and level flight), state updates drop to a predetermined rate (e.g., every five seconds)

• “Dead Reckoning” algorithms
  – extrapolate current position of moving objects based on last reported position

• Simulation time constraints
  – many simulations are human-in-the-loop
  – humans cannot distinguish temporal difference < 100 milliseconds
  – places constraints on communication latency of simulation platform
A Typical DVE Node Simulator

Execute every 1/30th of a second:
- receive incoming messages & user inputs, update state of remote vehicles
- update local display
- for each local vehicle
  - compute (integrate) new state over current time period
  - send messages (e.g., broadcast) indicating new state

Typical Sequence

1. Detect trigger press
2. Audio “fire” sound
3. Display muzzel flash
4. Send fire PDU
5. Display muzzel flash
6. Compute trajectory, display tracer
7. Display shell impact
8. Send detonation PDU
9. Display shell impact
10. Compute damage
11. Send Entity state PDU indicating damage
Summary

• Distributed Virtual Environments have different requirements compared to analytic simulations, leading to different solution approaches
  – May be acceptable to sacrifice accuracy to achieve better visual realism
  – Limits of human perception can often be exploited

• Distributed Interactive Simulation (DIS) representative of approach used in building DVEs