Collaborative Virtual Environments

071011-1
Fall 2020
10/28/2020
Kyoung Shin Park
Computer Engineering
Dankook University

What is Collaborative Virtual Environment?

- A software system in which multiple users interact with each other in real-time, even though those users may be physically located in distant places.
- Typically, each user accesses his/her own computer workstation or console, using it to provide a user interface to the content of a virtual environment.
- These environments usually aim to provide users with a sense of realism by incorporating realistic 3D graphics, spatial sound & other modalities to create an immersive experience.

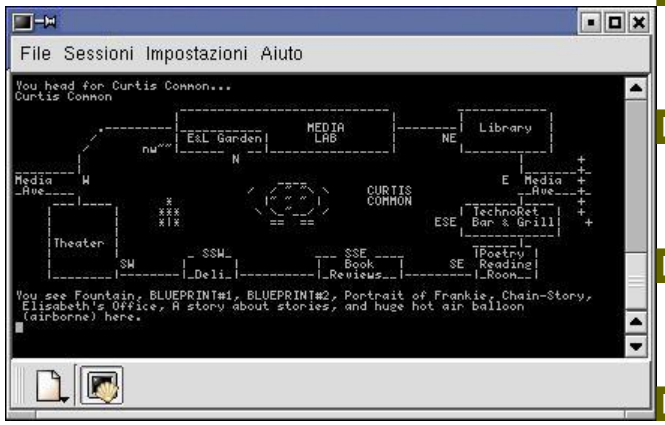
Characteristics of CVE

- Shared sense of space
- Shared sense of presence
- Shared sense of time
- A way to communicate
- A way to share

CVE Examples

- MUDs & MOOs starting in late 1970s
- Networked Games
  - DOOM, SGI Flight & Dogfight
- VRML-based Online Community
  - **Active World**, Blaxxun, Sony's Community Place, Open Community, Vnet , **Second Life (late 1990s and early 2000s)**
- **MMORPG (Massively Multiplayer Online Role Playing Games)** from early 2000s to the present
- Networked/Collaborative Virtual Environments
  - **SIMNET/Distributed Interactive Simulation (DIS)/NPSNET 1995**
  - DIVE
  - BrickNet
  - MR Toolkit
  - Diamond Park
  - CAVERNsoft

MUDs & MOOs



- MOO (MUD, Object-Oriented) & MUD (Multi-User Domain)
- Text-based virtual reality environment
- Originally designed as a form of the Dungeons and Dragons game
- Developed for multi-users on the Internet
- Allows users to interact both with their environment and with other users
- Descriptions of real and imagined areas such as forests, dungeons, offices, universities, cities, rooms, or any other spatially oriented environment
- Communication commands are modeled on real life, with “say”, “tell”, “whisper” and “shout”

Doom

- Dec 1993, id Software released its shareware game, **Doom**.
- Startup into the business of providing online gaming networks.
- The posting of Doom caught most network administrators' eyes when their LANs started bogging down. Doom flooded LANs with packets at frame rate.
- This networked ability to blast people in a believable 3D environment created enormous demand for further 3D networked games.
- An estimated 15 million shareware copies of Doom have been downloaded around the world, passed from player to player by floppy disk or online networks.

VRML-based Collaborative Virtual Environments

- Internet-based collaborative virtual environments will impact the greatest number of people.
- Examples are Active World, Blaxxun, Sony's Community Place, Open Community, Vnet
- Also, online gaming systems
- Problems of latency, rendering, inconsistency, lack of interaction

Second Life

'Dokdo is Korea Territory!' in Second Life
www.serakorea.com

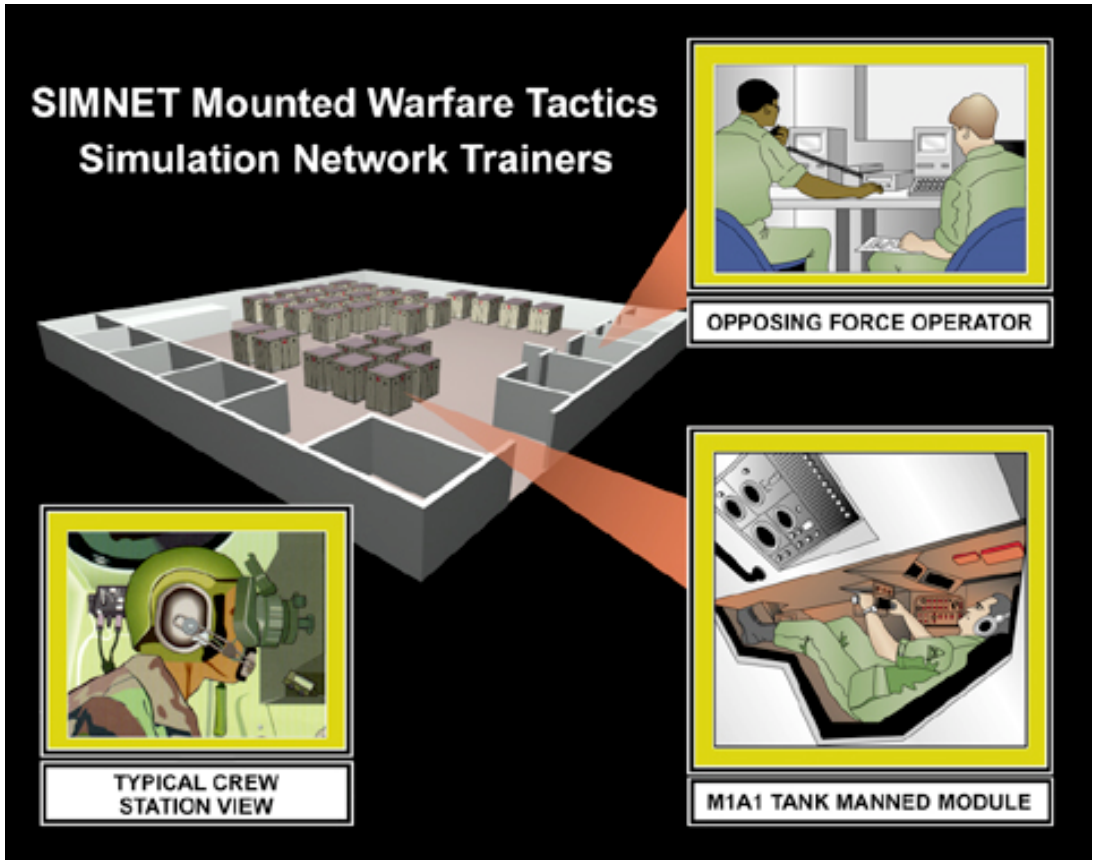
- Second Life is an online 3D virtual world community, developed by Linden Lab and modelled after the Metaverse of Snow Crash.

MMORPG

Ultima Online

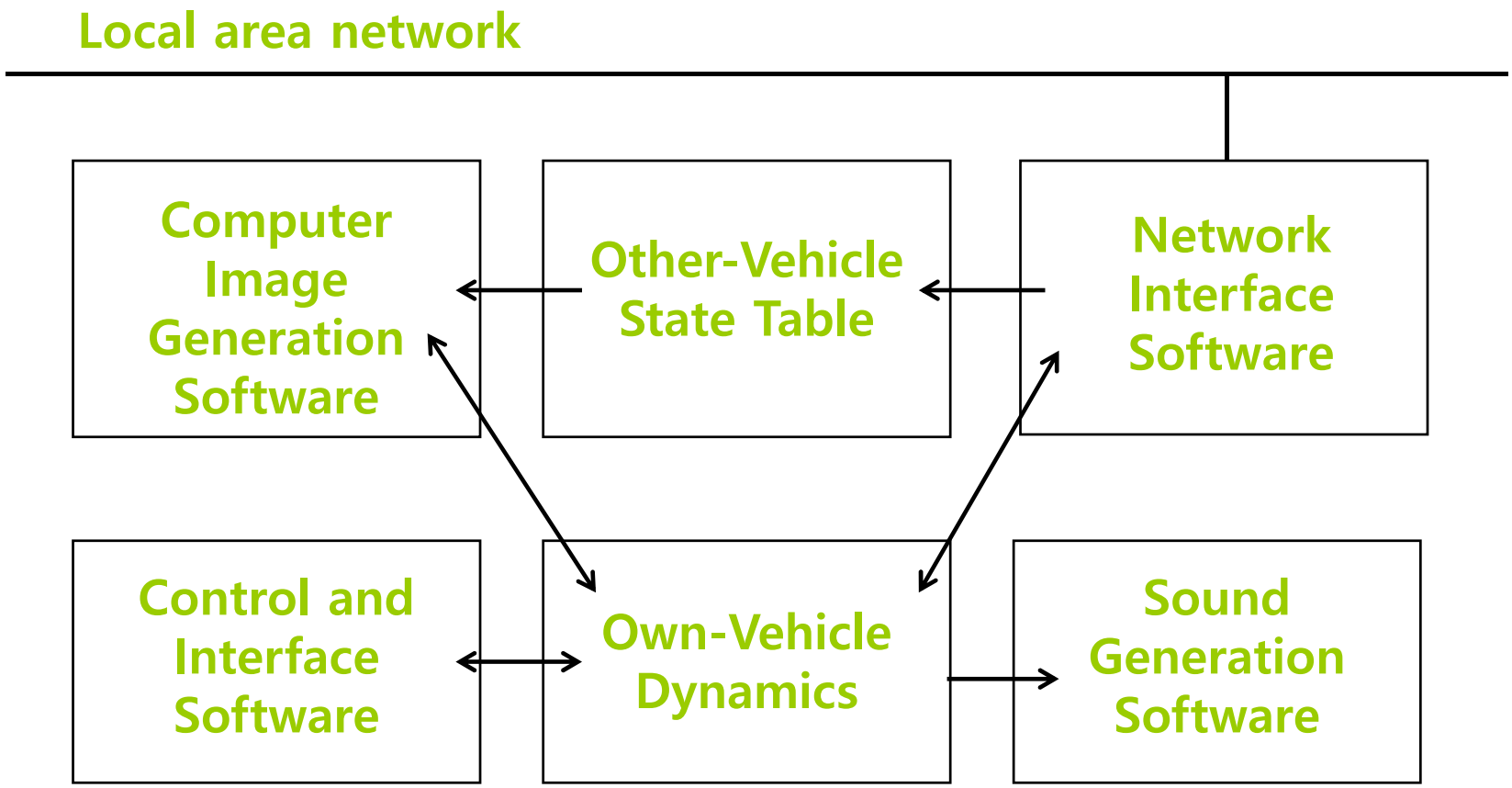
- Massively multiplayer online role-playing game (MMORPG) is a genre of computer role-playing games, in which a large number of players interact with one another in a virtual world.
- Richard Garriott, the creator of Ultima Online, coined the term MMORPG.
- Popular examples are Neverwinter Nights, Ultima Online, EverQuest, Blizzard's World of Warcraft.

SIMNET (Simulator Networking)



- SIMNET (simulator networking) is a distributed military virtual environment.
- The goal was to develop a "low-cost" networked virtual environment for training small units to fight as a team.
- SIMNET project created an 11-site testbed with from 50 to 100 simulators at each site.

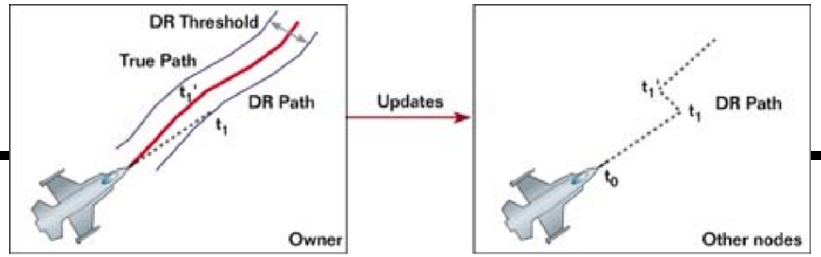
SIMNET Architecture



SIMNET Architecture

- The SIMNET network software architecture has three basic components
  - An **object-event architecture**
  - A notion of **autonomous simulation nodes**
  - An embedded set of predictive modeling algorithms called "**dead reckoning**"
- Object-event architecture
  - The world as a collection of objects whose interactions with each other are just a collection of events.
  - Objects are the vehicles and weapons systems that can interact across the network.
  - Events are messages indicating a change in world or object state.

SIMNET Architecture



- Autonomous simulation nodes
  - Individual players (vehicles and weapons) on the network are responsible for placing messages, or packets, onto the network to accurately represent their current state.
  - Packet recipients are responsible for receiving such state change information and making the appropriate changes to their local model of the world.
  - **Heartbeats:** usually every 5 seconds, to keep other players informed that a particular object is alive in the system.
- **Dead reckoning**
  - Objects only place packets onto the network when their home node determines that the other nodes on the network are no longer able to predict their state within a certain threshold amount.

SIMNET Scalability

- The SIMNET network software architecture proved scalable with an exercise in March of 1990 having some 850 objects (1 packet per sec) at five sites, with most of those objects being semi-automated forces.
- Objects in that test averaged one packet per second, with each packet being some 156 bytes in size for a peak requirement of 1.06 Mbits/second, just under the T-1 speed(1.544 Mbps) of the connecting links.

Distributed Interactive Simulation (DIS)

- Fully distributed heterogeneous network software architecture
- The environment can include virtual players (driven by a live human at a computer console of some sort), constructive players (computer-driven players), and live players (actual weapons systems plugged into the DIS network).
- The US Army's Close Combat Tactical Trainer (CCTT) is one of the larger scale networked virtual environments.

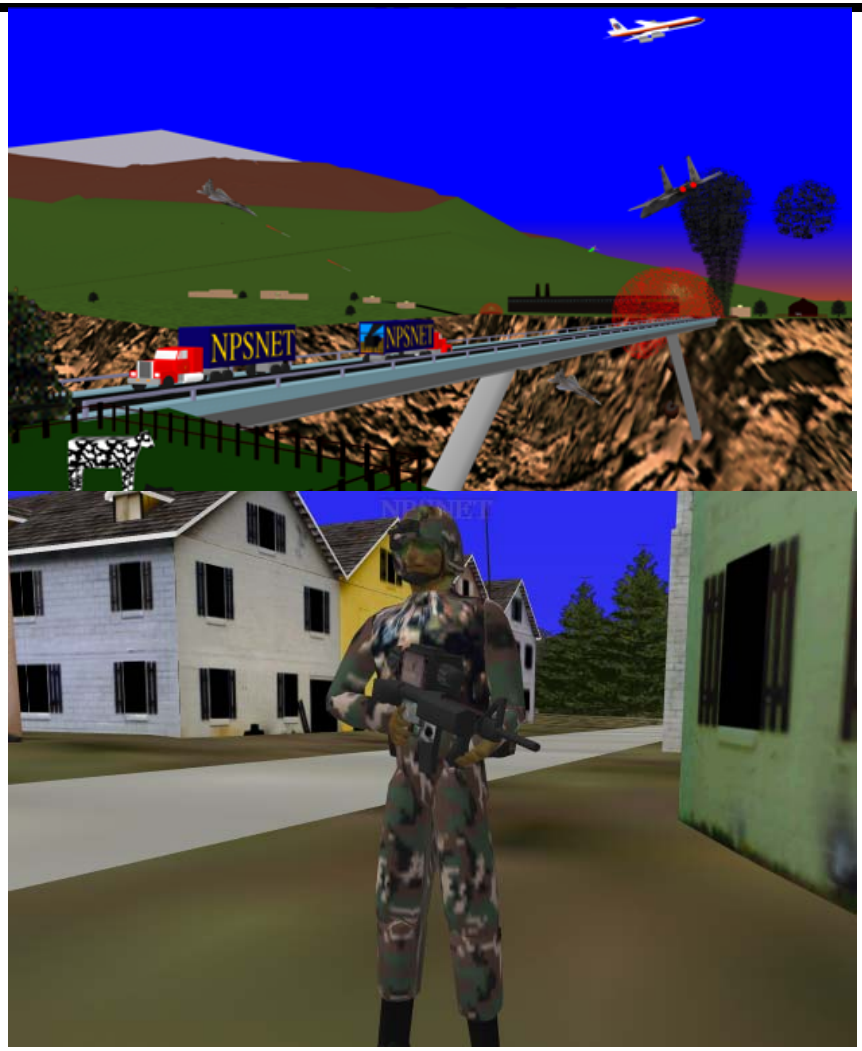
Distributed Interactive Simulation (DIS)

- DIS is a direct descendent from SIMNET but has packets that are more general than SIMNET's.
- DIS has three basic components
  - Object-event architecture
  - Notion of fully distributed simulation nodes
  - Embedded set of predictive modeling algorithms for "dead reckoning"
- The core of the DIS network software architecture is the data sharing via Protocol Data Unit (PDU).
- The DIS (IEEE 1278) standard defines *27 different PDUs*, only four of which (Entity State, Fire, Detonation, and Collision) are used by nodes to interact with the virtual environment.
  - A demonstration at the 1993 showed that Entity State PDUs comprised 96% of the total DIS traffic.
  - Remaining 4% distributed mainly amongst Transmitter (50%), Emission (39%), Fire (4%), and Detonation (4%).

NPSNET

- To implement a large-scale networked virtual environment
- NPSNET-1,2&3
  - NPSNET-1 was demonstrated live at the SIGGRAPH 91
  - NPSNET-1 did not use dead-reckoning. NPSNET-1 flooded the network with packets at frame rate.
  - NPSNET-2 and 3 were utilized to explore better, faster ways to do graphics, and to extend the size of the terrain databases possible.
- NPSNET IV
  - NPSNET-IV was DIS-compliant, dead-reckoned and had spatial sound.
  - NPSNET-IV has interoperated with almost every DIS-compliant virtual environment ever constructed.
- NPSNET-IV Capabilities
  - Building walkthroughs, Articulated humans, Networking - play across the multicast backbone of Internet.

NPSNET-IV



NPSNET-IV

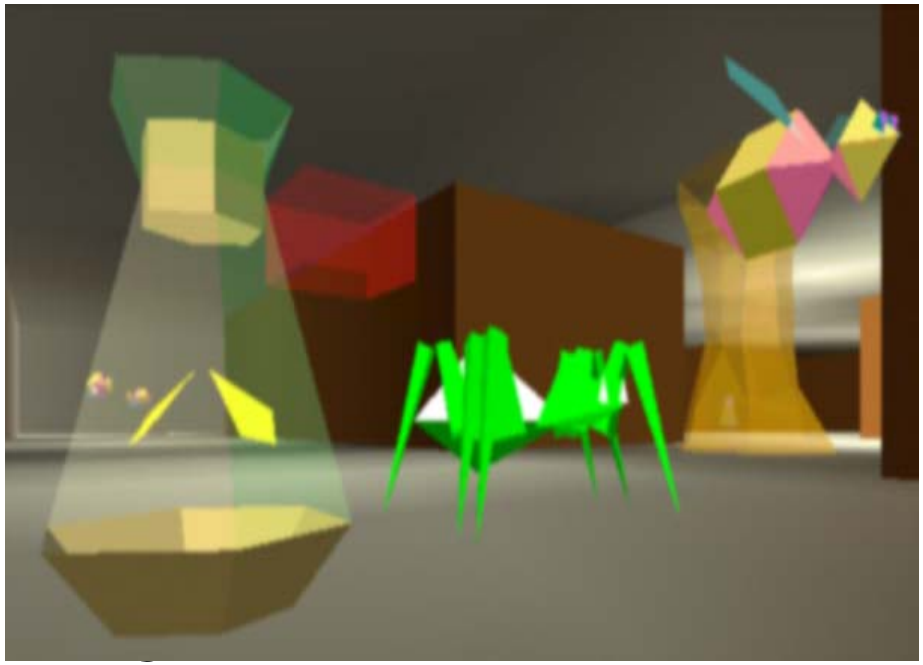
DIVE

- The Swedish Institute of Computer Science's Distributed Interactive Virtual Environment (DIVE) is another early and ongoing academic collaborative virtual environment.
- DIVE has a homogeneous distributed database like SIMNET and DIS-compliant systems.
- Unlike SIMNET, the entire database is dynamic and uses **reliable multicast protocols** to actively replicate new objects.

DIVE

- A disadvantage with this approach is that it is difficult to scale-up because of the communications costs associated with maintaining reliability and consistent data.
- For example, modeling terrain interactions, such as building a bern, still would be very expensive (though highly desirable) in terms of the number of polygons that would need to be created, changed, and communicated in DIVE.

BrickNet



- BrickNet is developed by the Institute of Systems Science at the National University of Singapore.
- A client-server model in which the database is partitioned among clients.
- Communication is mediated by central servers.
- For example, as an entity moves through the VE, its database is updated by an object-request broker on a server that has knowledge of which client maintains that part of the world.

Diamond Park

https://www.youtube.com/watch_popup?v=duzn3ULqS-A

- Mitsubishi Diamond Park has multiple users that interact in the park by riding around on bicycles and talking to each other (Social VR)
- The MERL Diamond Park VE is built using SPLINE (Scalable PLatform for INteractive Environments) which provides the implementation of **locales & beacons**.

Diamond Park/SPLINE

- **Locales** are an efficient method for managing the flow of data between large numbers of users in a large-scale VE
- The concept of locales is based on the idea that while a VE may be very large, most of what can be observed by a single user at a given moment is local in nature.
- Each locale is associated with a separate communication channel, and each locale has its own coordinate system.
- **Beacons** are a special class of objects that can be located without knowing what locale they are in (to solve the "how do I join the VE problem").
- Beacons act as a content-addressable index from tags to the multicast address of locales. They make it possible to decide what locales to attend to based on what the locales contain.

CAVERNsoft/QUANTA

CAVERNsoft/QUANTA

- C++ toolkit for building Tele-Immersive applications with special emphasis on networking
- Client-server topology
- Higher-level networking and database APIs and tools for application developer modules
- Available for Windows, SGI IRIX, Linux, FreeBSD, Sun Solaris, HP Unix, WinCE
- Graphics support for IRIS Performer

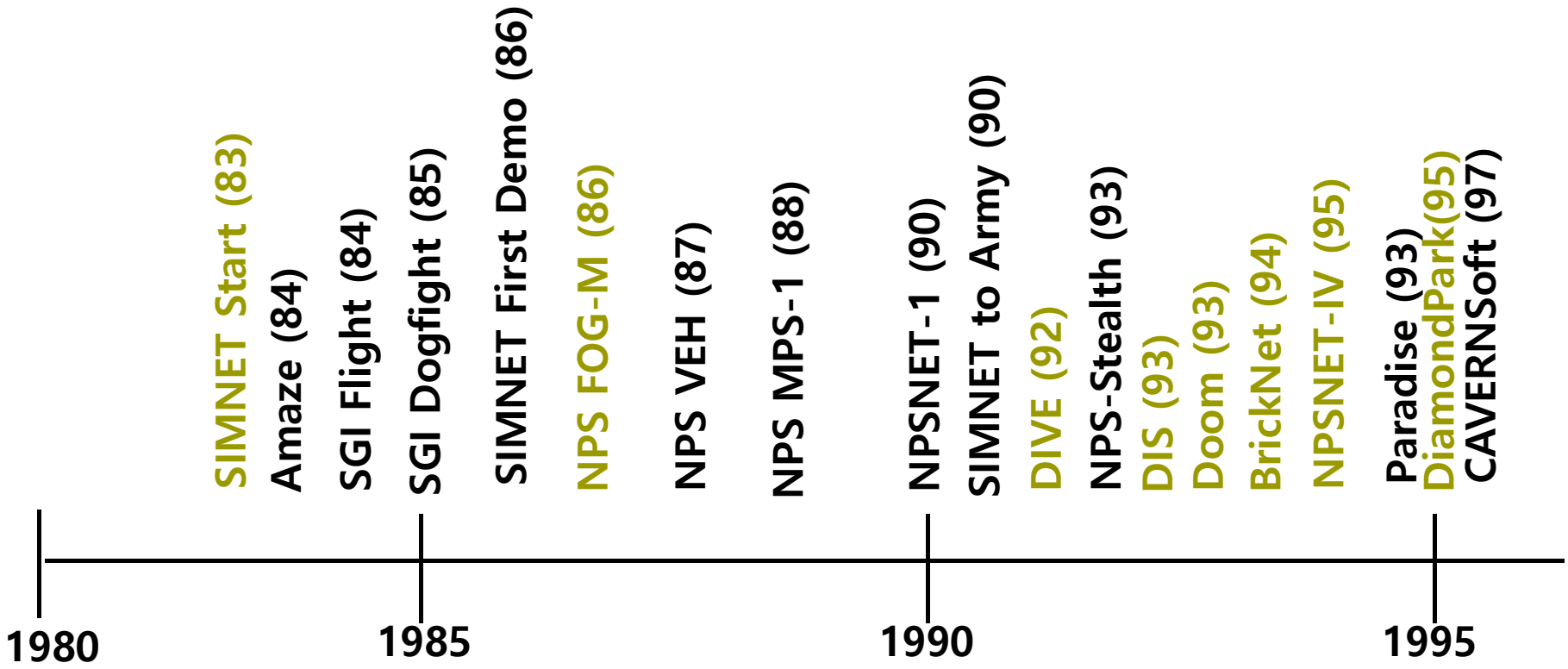
Low-Level Components

- Most of these capabilities have demo programs
- TCP, UDP, multicast, HTTP
- UDP reflector and multicast bridge
- TCP reflector
- Remote procedure calling (RPC)
- Remote File I/O
- Client/Server Databases
- Parallel Socket TCP
- Reliable Blast UDP (RBUDP)
- Cross-platform Data Conversions
- Mutual exclusion and threading
- Performance Monitoring- Netlogger compatible
- Implemented across SGI, Windows9x/NT/2000, Linux, FreeBSD

High-Level Developer Modules

- Audio streaming
- Base and Articulated avatars
- VR navigation and collision detection
- VR picking and moving
- VR network dynamic coordinate system
- VR menus
- Speech recognition with IBM ViaVoice
- Collaborative Animator
- Collaborative application shell to jumpstart development

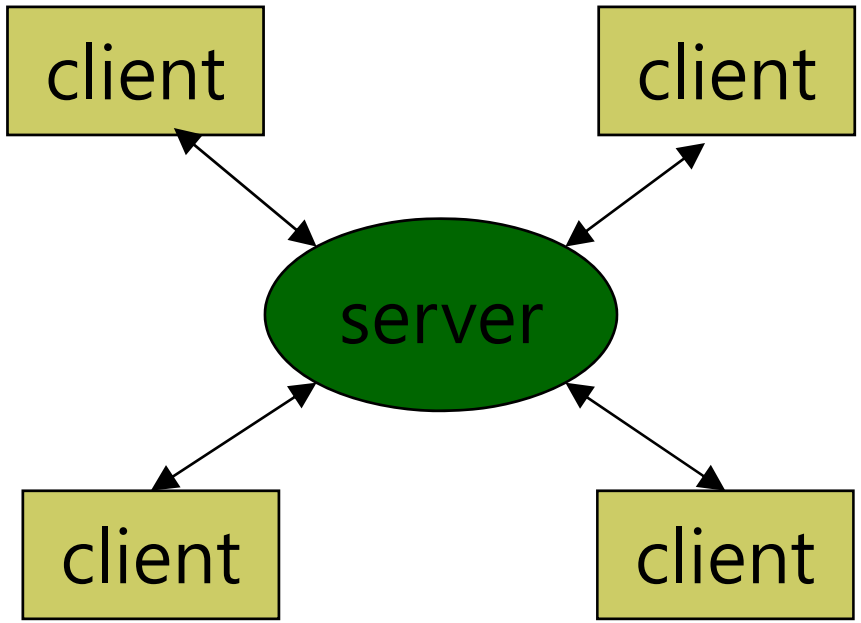
A Brief Timeline of Networked-VEs



Challenges of CVE Design & Developments

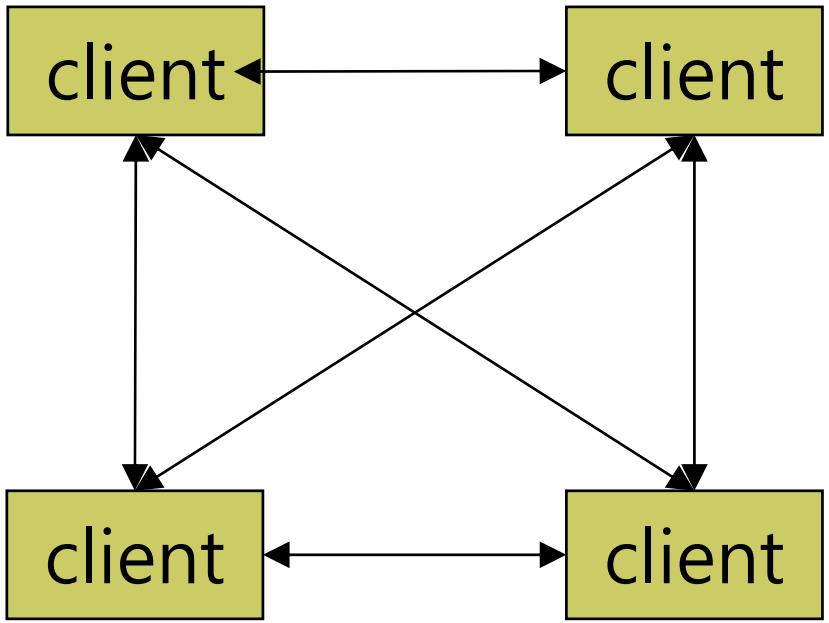
- Network Bandwidth
- Heterogeneity
- Distributed Interaction
- Real-time System Design and Resource Management
- Failure Management
- Scalability
- Deployment and Configuration

Network Model: Centralized



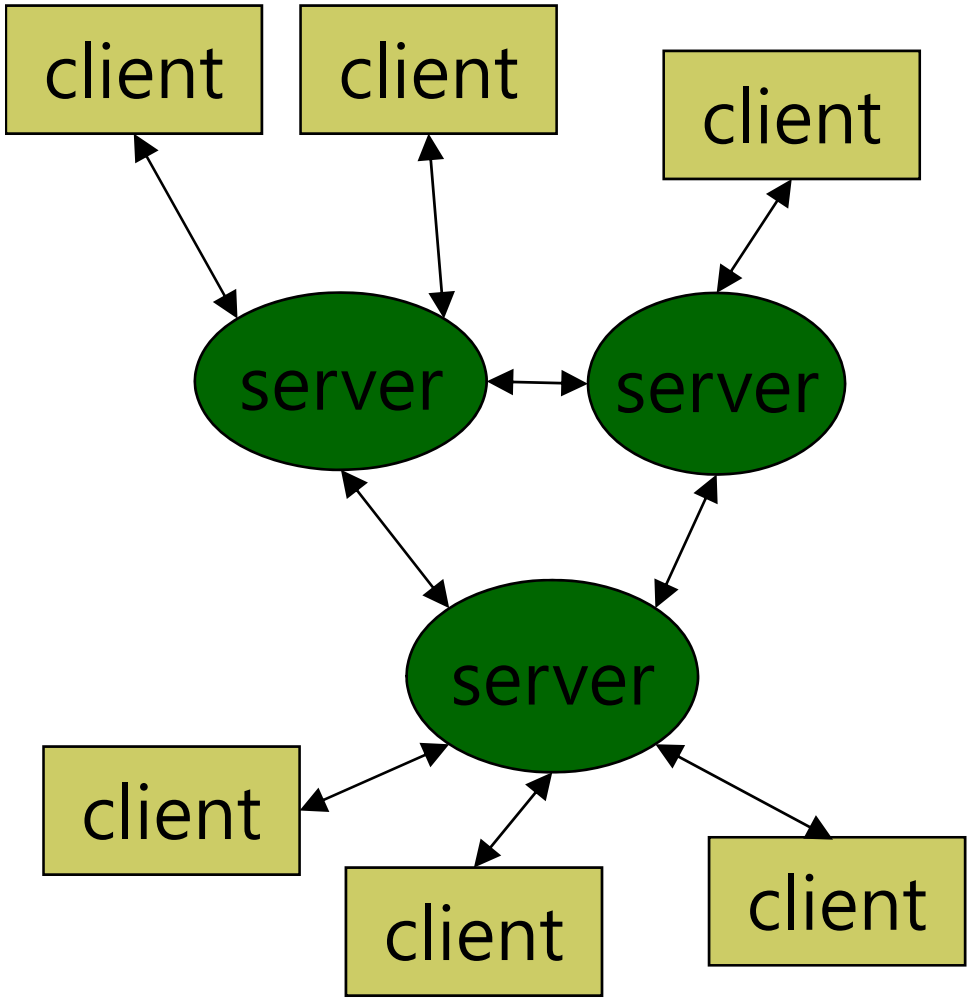
- Client-server model
- One computer (server) collects all data and sends updates to the users (clients)
- Simple structure, easy to maintain database (useful for compression & admin tasks)
- Not scalable, the central server is the bottleneck

Network Model: Distributed



- Peer-to-peer model
- Each user maintains its own copy of the database
- Updates are send to other users
- Difficult to manage the number of connections
- Not scalable, the network is the bottleneck

Network Model: Hybrid

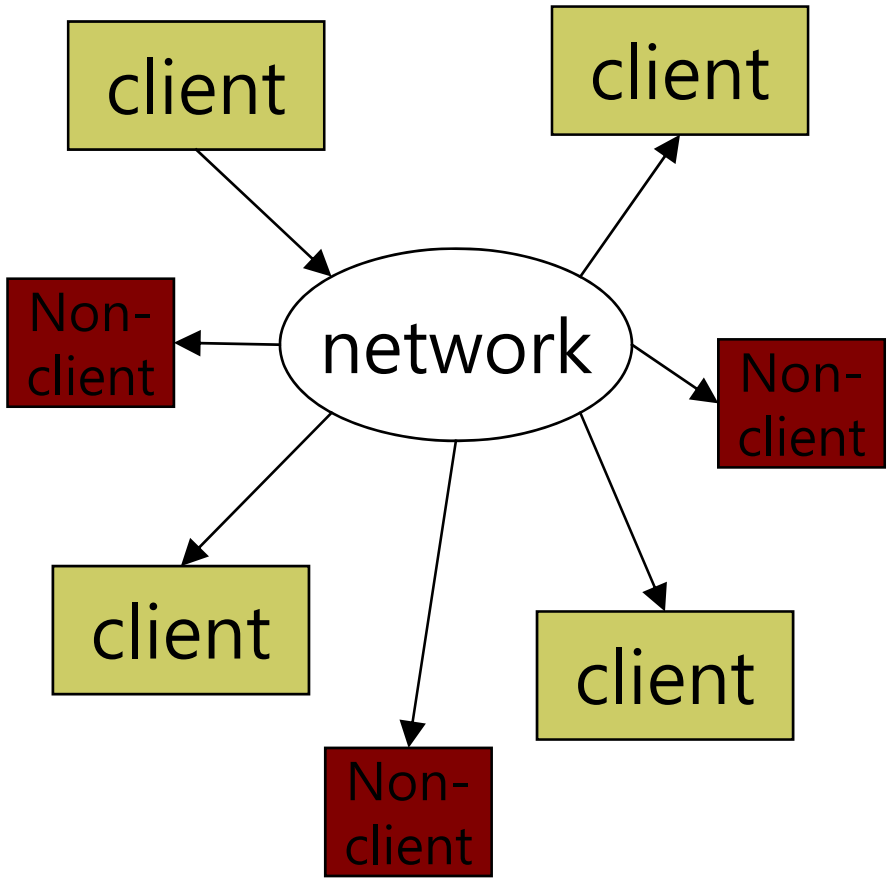


- Multi-players client-server model with multiple servers
- If we are using a multiplayer videogame service company

How to avoid bottlenecks?

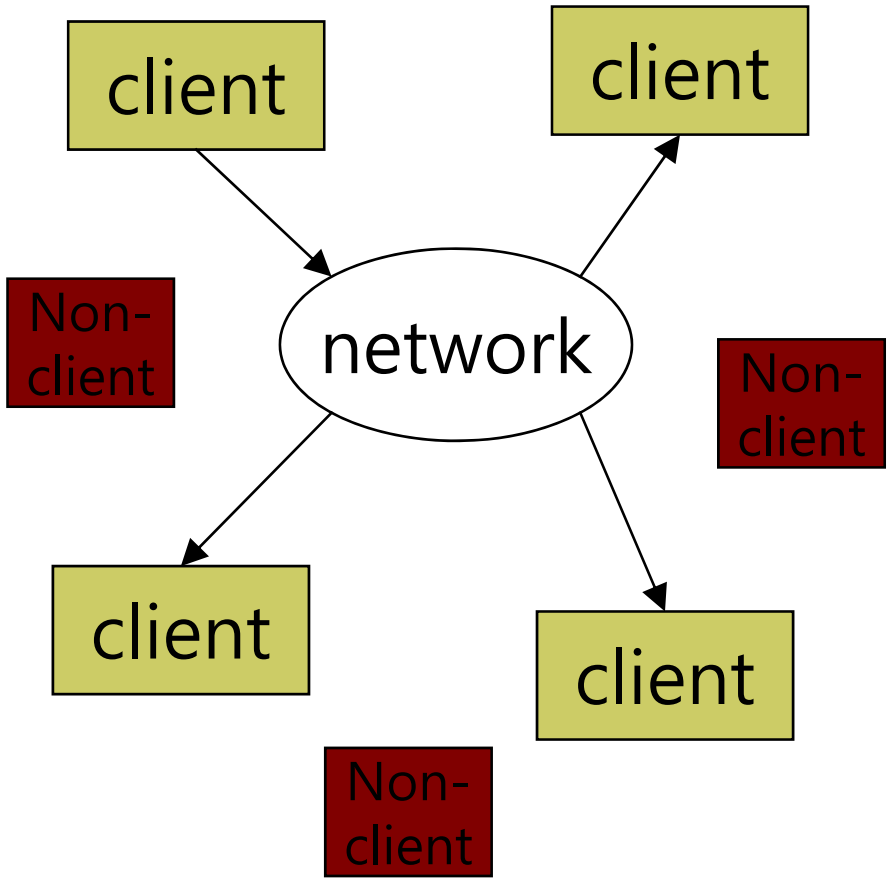
- Better communication models
  - Reduce number of connections and messages
- Better database models
  - Distributed databases
- Better decision making
  - Make it distributed but any given decision is made in only one place

Broadcast Communication



- The message is sent to all users (and non-users)
- Not selective
- Floods network with packets
- All packets must be brought up through the kernel of the operating systems of all users
- Even if the packet is not for that machine! Thereby, wasting CPU time.

Multicast Communication

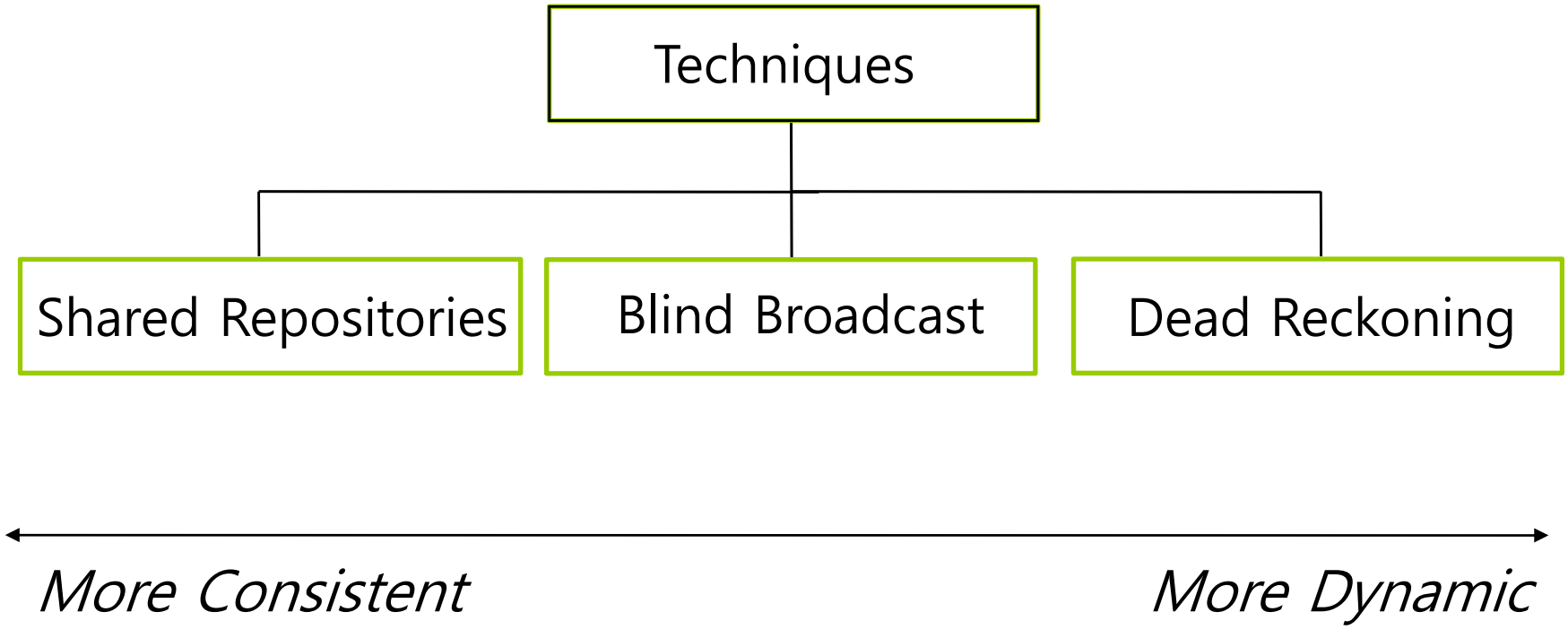


- The message is sent to the multicast group (and therefore to all group members)
- Non-users (non-group users) do not receive messages
- Multicast services allow arbitrarily sized groups to communicate on a network via a single transmission by the source.
- Can inter-network (route over the network layer) with multicast.

What is Dynamic Shared State?

- The dynamic information that multiple hosts maintain CVE
- Accurate dynamic shared state is fundamental to creating realistic virtual environments. It is what makes a VE "multi-user".
- Management is one of the most difficult challenges facing the CVE designer. The trade off is between resources and realism.
- Network latency problem
- For a highly dynamic shared state, hosts must transmit more frequent data updates.
- To guarantee **consistent views** of the shared state, hosts must employ reliable data delivery.

Managing Shared States



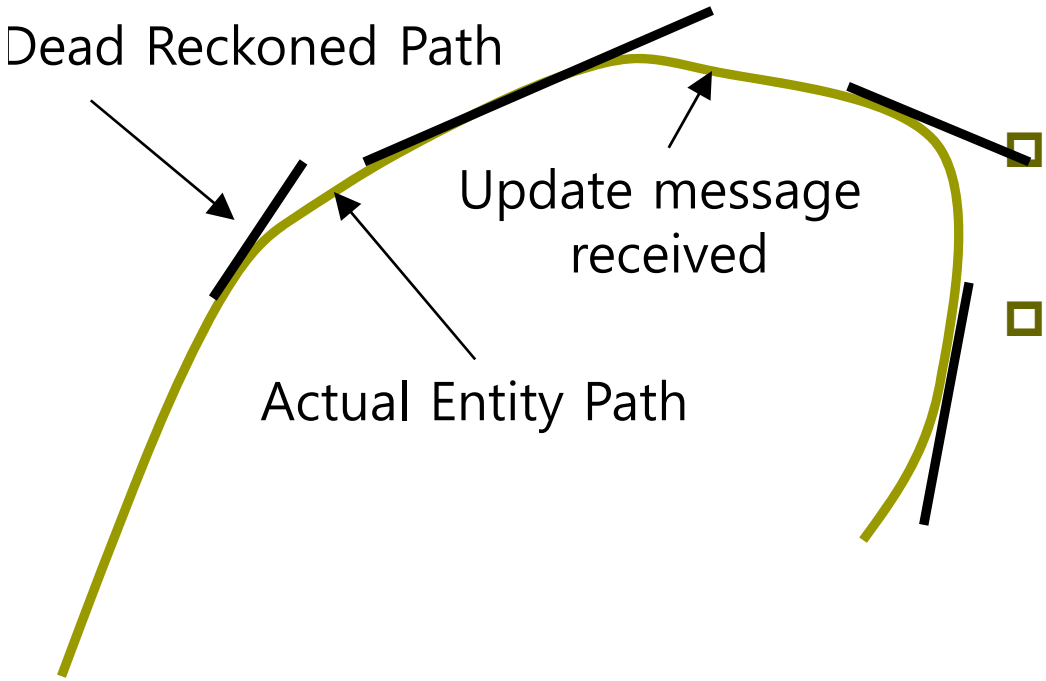
Shared Repository

- Maintain shared state data in a centralized location.
- Protect shared states via lock manager to ensure ordered writes.
- Shared File Directory
  - Absolute Consistency!
  - Only one host can write data to the same file at a time. Must have locks. Hence, does not support many users.
- Server Memory
  - Faster than Shared file because each host uses does not have to open and close each file remotely.
  - Don't have to have locks. Server arbitrates. Server crash is catastrophic
  - Maintaining constant connection may strain server resources.
- Virtual Repository
  - Tries to reduce bottleneck at server. Better fault tolerance.

Frequent State Regeneration/Blind Broadcasts

- Owner of each state transmits the current value asynchronously and unreliably at regular intervals.
- Clients cache the most recent update for each piece of the shared state.
- Hopefully, frequent state update compensate for lost packets.
- Broadcast is sent “blind” to everyone.
- No acknowledgements, No assurances of delivery, No ordering of updates.
- Used where it may not have demanding consistency requirements.
- Each host takes explicit ownership of one piece of the shared state (usually the user’s avatar).
- Commonly used in online game (Doom, Diablo)

Dead-reckoning



- The objects and ghosts paradigm
- An algorithms to reduce number of messages
- Instead of sending frequent updates on object's position, it is calculated locally using a last-known **velocity and position**
- No need for central server
- Sacrifices accuracy of shared state for more participants

Dead-reckoning

- Implementation:
  - Every user has a copy of the database
  - Each user is in charge of moving all of the objects within its database
    - Direct control ("live" object)
    - Dead-reckoning ("ghost" object)
  - Dead-reckoning is used on "live" objects when difference from direct control is significant, updates are sent.
- Characteristics:
  - Reduces bandwidth
  - Live and ghost objects have different update rates, prediction and convergence needed (I.e. no guarantee that all users have identical state about each object)
  - Requires customization based on object behavior

Dead Reckoning

- Advantages
  - Reduces bandwidth requirements because updates are sent less frequently.
  - Potentially larger number of players.
  - Each host does independent calculations
- Disadvantages:
  - Not all hosts share the identical state about each entity.
  - Protocols are more complex to implement to develop, maintain and evaluate.
  - Must customize for object behavior to achieve best results.
  - Must have convergence to cover prediction errors.
  - Collision detection difficult to implement.
  - Poor convergence methods lead to jerky movements and distract from immersion.

Heartbeats

- Each user periodically sends a message called a **heartbeat** informing everyone of its status.
- Usually every 5 seconds, to keep other players informed that a particular object is alive and still in the system (and hence should be displayed).
- Entities must have a “heartbeat” otherwise cannot distinguish between live entities and ones that have left the system.
- Helps recovery from lost messages (to help network reliability)
- Helps users who just joined

Real-time Rendering Challenges

- Real-time rendering
  - Polygon culling & Level-of-Detail processing
- Real-time collision detection and response
  - Who determines collision in a networked virtual environment?
- Computational resource management
- Interaction management

Polygon Culling and LOD Processing

- Try to use available CPU cycles to throw away most of our 3D model before we send it through the graphics pipeline.
- But we are about to get graphics engines that run over 100M polygons per second, some planning beyond 300M+ polygons per second, so maybe this becomes less of a problem.
- Use a hierarchical data structure for the displayable world.
- Create LOD models by hand in our modeling tool, throwing away small polygons for the low resolution versions of our models.
- Some modeling tools will do LOD semi-automatically. They give you a cut at it and you can add polygons back in.

Real-time Collision Detection and Response

- Movement through our CVEs requires that we have some way to determine if we have collided with the surfaces in our world so that we can stop our movement or react to the collision.
- Interactivity in our CVEs requires that we have some way of reaching out and touching an object in our VE, being able to determine what we touched and then being able to react.
- No matter how good the graphics and textures look, the poor realism resulting from a lack of collision detection breaks the suspension of disbelief.
- Systems targeted toward large-scale, interactive simulation environments include I-COLLIDE, RAPID, and V-COLLIDE.

Computational Resource Management

- Network bandwidth increases as the number of new users increases.
  - New users increase amount of shared data and level of interaction in the environment.
  - More network bandwidth is required to maintain the data and disseminate the interactions.
- As more users enter the CVE, additional processor cycles are required at each of the existing users' hosts.
  - Since each user introduces new shared state to the CVE, the processor must cache this additional state, receive updates to this new state, and apply those updates to the cache.
  - Because each user introduces additional updates, the processor must be prepared to receive and handle the increased volume of updates and support increased interactions with the local user.

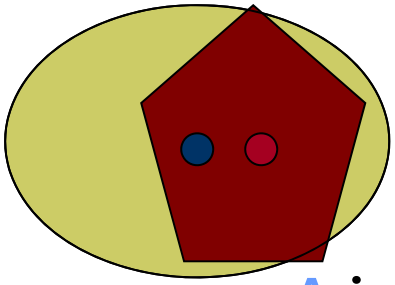
Computational Resource Management

- Communication protocol optimization
  - Reduce packet size and the number of packets
  - Packet compression – may be lossless or lossy and can be internal or external.
  - Aggregation – reducing the number of packets that are actually transmitted by merging information from multiple packets into a single packet.
- Data flow restriction
  - Controlling the visibility of data
  - Data flow management using **Area-Of-Interest** Management
- Leveraging limited user perception
- Modifying system architecture

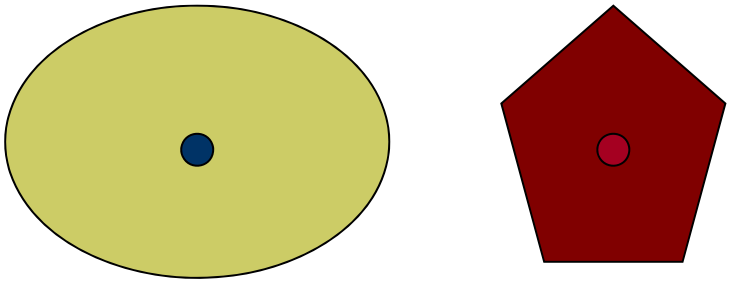
Data Visibility

- Resource management for scalability and performance
- The goal is to send information to those hosts who really need to receive it
- Individual user needs to know only a small portion of the total available information
- **Aura-Nimbus** approach
- **Area-Of-Interest** filter

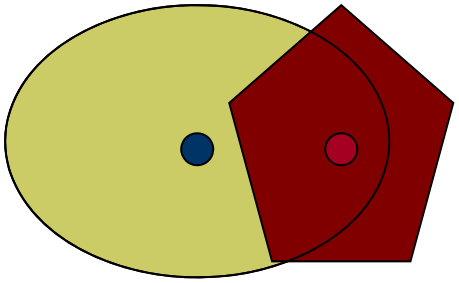
Aura-Nimbus Spatial Model of Interaction



A is fully aware of B



A is not aware of B



A is semi-aware of B

- Aura – data should only be available to those entities that are capable of perceiving that information
- Nimbus – data should only be available to those who are interested in that information
- Aura-Nimbus has the disadvantage in that it does not scale to large numbers of entities.
- Each packet has a custom set of destination entities – hard to utilize multicasting

Area of Interest Management (AOIM)

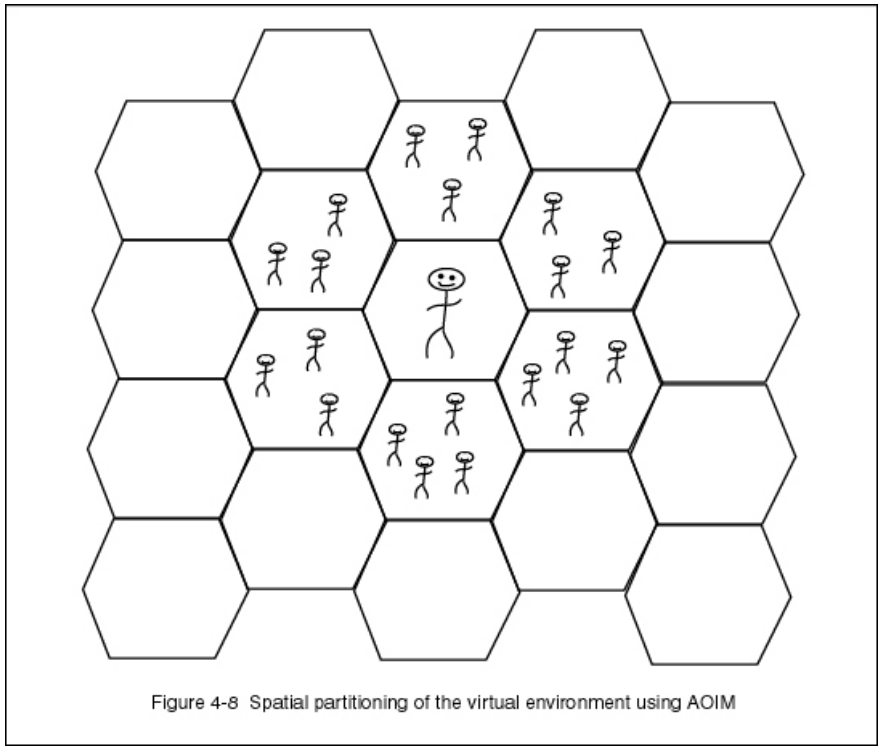


Figure 4-8 Spatial partitioning of the virtual environment using AOIM

- In the real world, which virtual environments emulate, entities have a limited "areas of interest".
- *Area of Interest filters* are explicit data filters provided by each host, allowing the CVE to perform fine-grained data management to deliver only the information the host needs.
- Or, multicasting network to restrict data flow
- Spatial, temporal, functional partitioning classes

Area of Interest Management (AOIM)

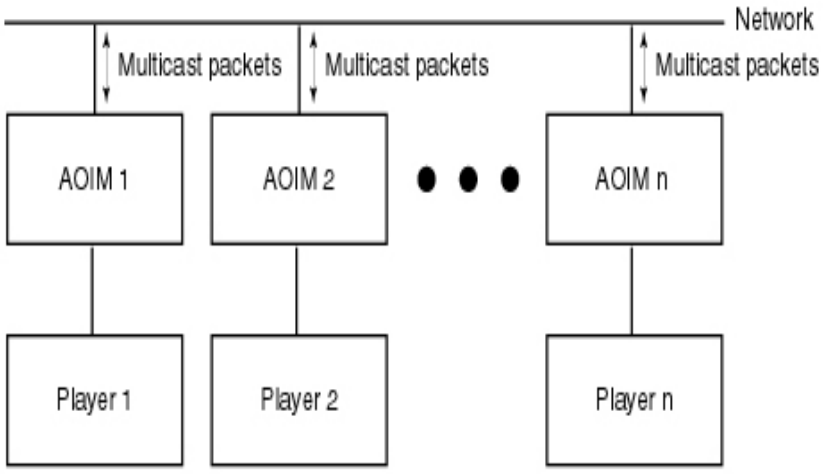


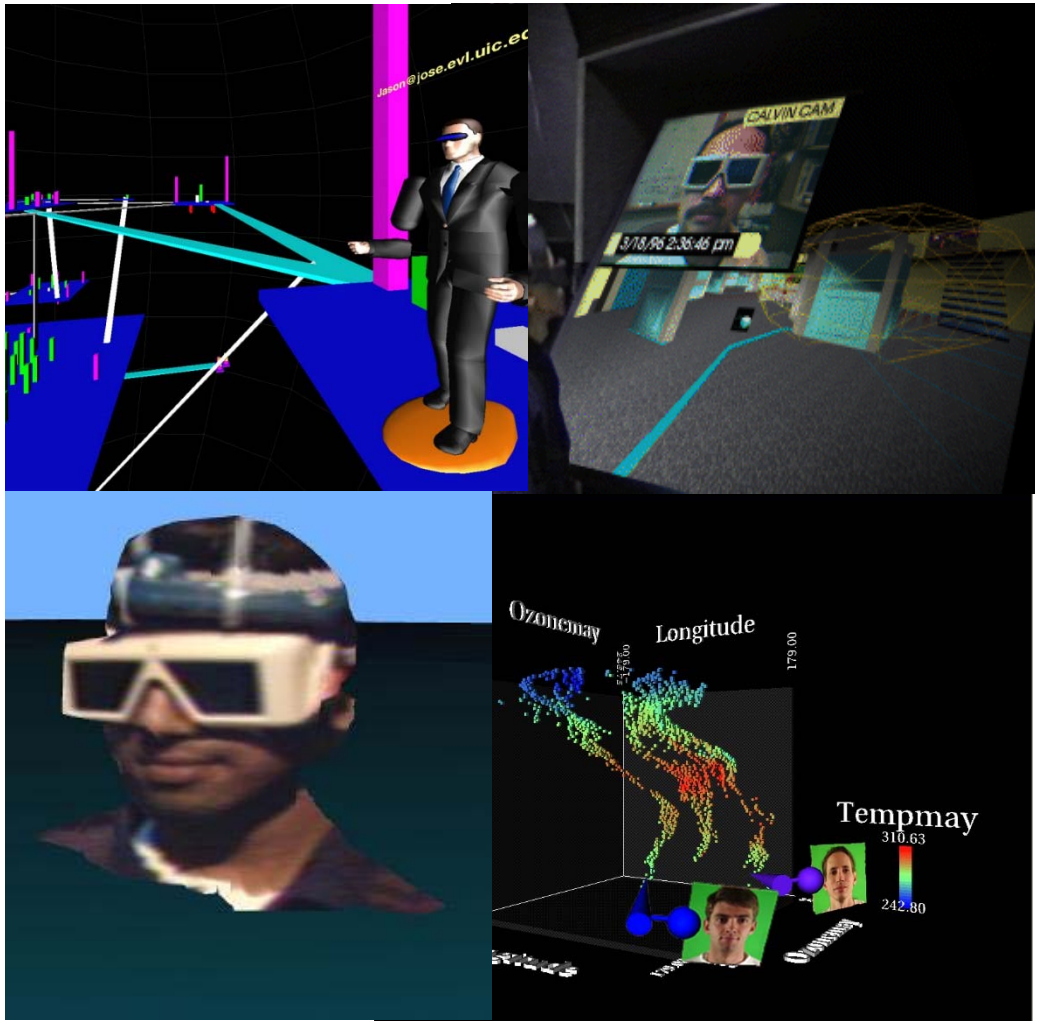
Figure 4-7 Area of interest management (AOIM) software layer

- Interactions are mediated by an AOIM software layer.
- Partition the simulation into workable chunks to reduce computational load on hosts and minimize communications on tail links.
- Distribute partitioning algorithms among hosts.

Collaboration Challenges

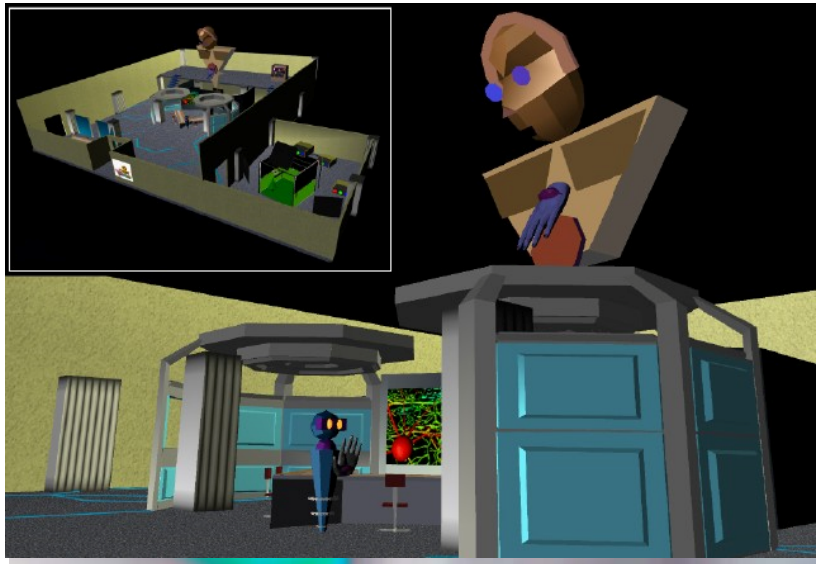
- "Natural, spatial" human-human communication
- Peripheral awareness
- Unification of communication and information
- Large number of participants
- Cooperative interaction

Avatar



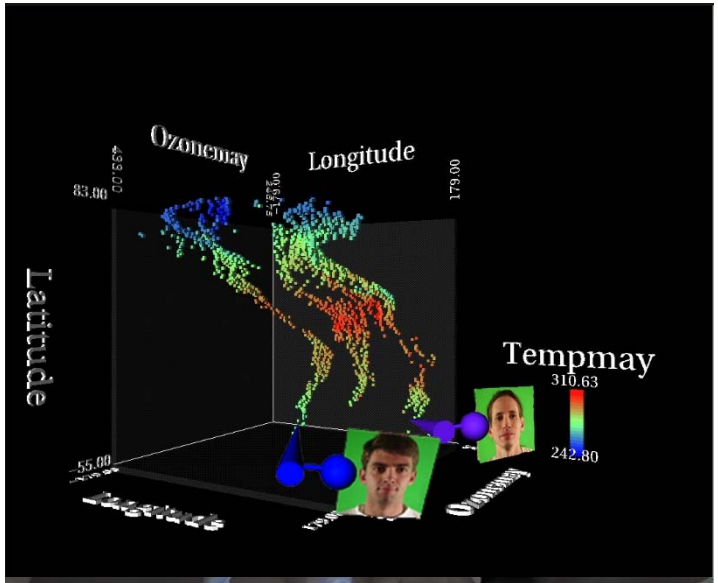
- Tracking head and hand position and orientation give good cues
- Pointing rays can be useful in large spaces
- Articulated avatars
- Pointers with static photographs attached
- Video as a window
- Video avatar

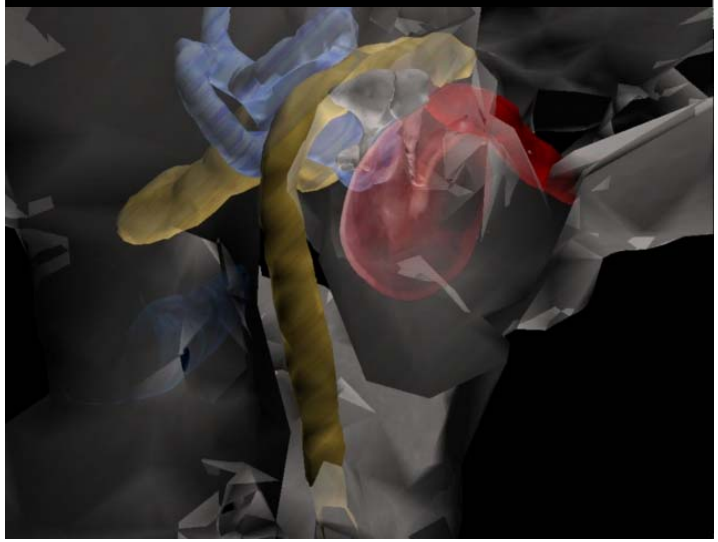
Avatar



- CALVIN - **simple articulated avatars** were used in this design environment that encouraged people to work at different scales to set up a configurable room.
- NICE - **more articulated avatars** in an educational setting, and usage by more casual users (issues of being able to see yourself, issues of 'equality', emergent social patterns)

Avatar





- TIDE - **pointers with static photographs** attached. One issue with pointers (and avatars) is knowing who is who. Attaching a name to a pointer is one solution where you can quickly talk to the person with the appropriate pointer.
- Virtual Temporal Bone - full body avatars would get in the way of this educational environment so only **different colored pointers** are used showing where the user's tracked wand is in the space.

Avatar

- Today you can also get yourself scanned in a 3D scanner and generate an **articulated avatar** of yourself
- https://www.youtube.com/watch_popup?v=DYlIOiFmwdc

Asynchronous Work - VR Annotator

- Sometimes asynchronous collaboration is better.
- VR annotations are recordings in VR where both the person's hand and head gestures as well as their voice is captured.
- Similar to attaching post-it notes to Adobe Acrobat files.
- **VR annotations** could be used to create virtual tour guides since the annotations are animated.

Some current consumer collaborative applications

- Cooperative games
  - Black Hat Cooperative
  - Keep Talking and Nobody Explodes
  - Bridge Crew
  - Rec Room
- Social VR
  - AltSpaceVR
    https://www.youtube.com/watch_popup?v=0I6QNXR0dPY
  - Facebook Spaces
    https://www.youtube.com/watch_popup?v=_kGRpSd4vnc

Reference

- Singhal, S. & Zyda, M., Networked Virtual Environments: Design and Implementation, Addison-Wesley, 1999
- Gossweiler, R. et al., An Introductory Tutorial for Developing Multiuser Virtual Environments, Presence 3(4), 1994, pp. 255-264.
- Carlsson C. & Hagsand, O. DIVE - A platform for multi-user virtual environments, Computers & Graphics, 17(6), 663-669, 1993.
- Leigh, J., Johnson, A., Brown, M., Sandin, D., DeFanti, T., Visualization in Teleimmersive Environments. In IEEE Computer, December, 1999, pp. 66-73