Special Course on

Networked Virtual Environments

Jouni Smed

Turku Centre for Computer Science (TUCS)

Department of Information Technology,
University of Turku

2004

Course Syllabus

- credits: 2 cu
- prerequisites: ‘knowledge on the basic concepts of computer networks’
- teaching methods: lectures (24 h)
  - Thursdays 8–10 and Fridays 8–10, Auditorium
  - from January 22 to February 27
- assessment: examination
- course web page:
  `http://staff.cs.utu.fi/staff/jouni.smed/scnve/`

Examinations 1 (2)

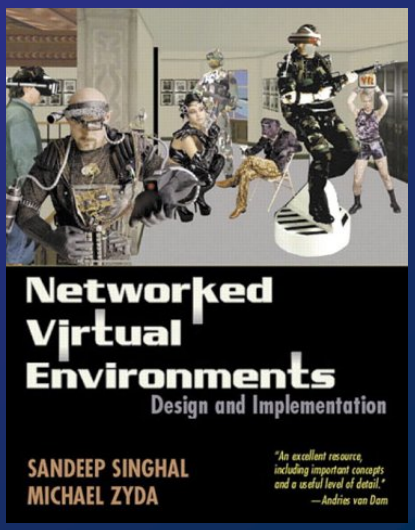
- examination dates
  1. March 15, 2004
  2. April 5, 2004
  3. May 10, 2004
- check the exact times and places at `http://www.it.utu.fi/opetus/tentit/`
- if you are *not* a student of University of Turku, you must register to receive the credits
  - further instructions are available at `http://www.tucs.fi/Education/Information/regcredits.php`

Examinations 2 (2)

- questions
  - based on the lectures and additional literature (3 articles)
  - four questions à 8 points
  - to pass the examination, at least 16 points (50%) are required
  - questions are in English, but you can answer in English or in Finnish
- remember to enrol in time!

Course Book

- S. Singhal and M. Zyda, *Networked Virtual Environments: Design and Implementation*, Addison-Wesley, Reading, MA, 1999.
- Chapters 1–7 (pp. 1–249)



Additional Literature

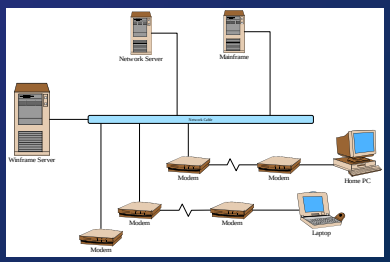
- S. Singhal, *Effective Remote Modeling in Large-Scale Distributed Simulation and Visualization Environments*, PhD thesis, Stanford University, Stanford, CA, 1996. Chapter 2, pp. 13–33.
- S. Benford, C. Greenhalgh, T. Rodden, and J. Pycock, Collaborative virtual environments, *Communications of the ACM*, 44(7):79–85, 2001.
- J. Smed, T. Kaukoranta, and H. Hakonen, Aspects of networking in multiplayer computer games, *The Electronic Library*, 20(2):87–97, 2002.

Outline of the Course

1. Introduction
2. Background
   - history
   - past projects and applications
3. Networking
   - data transfer and protocols
   - communication architectures
4. Managing dynamic shared state
   - consistency-throughput trade-off
   - centralized information repositories
   - frequent state regeneration
   - dead reckoning
5. System design
   - threads
   - polygon culling and level-of-detail
6. Resource management
   - packet compression and aggregation
   - area-of-interest filtering
   - exploiting perceptual limitations
7. Other issues
   - security
   - case examples

§1 Introduction

- Networked Virtual Environment (NVE) 'is a software system in which multiple users interact with each other in real-time, even though those users may be located around the world.'
  —Singhal & Zyda, 1999
- Keywords:
  - global
  - real-time
  - multiple
  - user
  - interaction



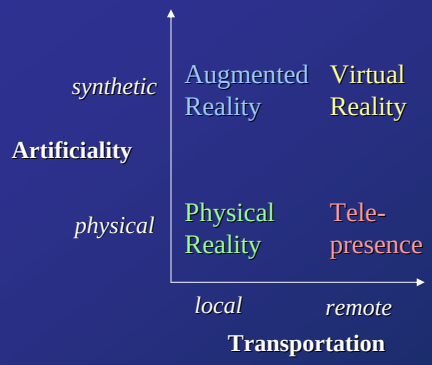
Application Areas for NVEs

- Military and industrial team training
- Collaborative design and engineering
- Multiplayer games
- Mobile entertainment
- Virtual shopping malls
- Online tradeshows and conferences
- Remote customer support
- Distance learning

Synonyms, Keywords and Abbreviations

- Collaborative Virtual Environment (CVE)
- Computer-Supported Co-operative Work (CSCW)
- Media-spaces, shared spaces
- Distributed Interactive Simulation (DIS)
- Distributed Virtual Environment (DVE)
- Virtual Reality (VR), Virtual Environment (VE), Virtual Worlds
- Augmented Reality (AR)
- ...

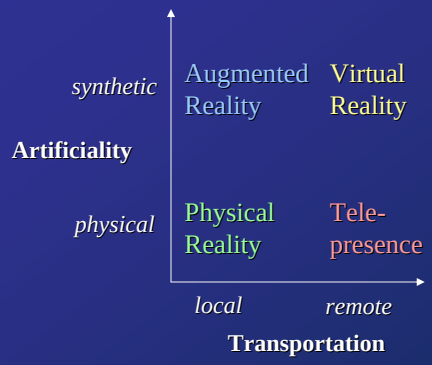
Classification of Shared-Space Technologies 1 (2)



- Physical reality
  - resides in the local, physical world
  - here and now
- Telepresence
  - a real world location remote from the participant's physical location
  - a remote-controlled robot

Benford et al., 1998

Classification of Shared-Space Technologies 2 (2)

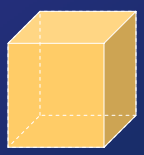


- Augmented reality
  - synthetic objects are overlaid on the local environment
  - a head-up display (HUD)
- Virtual reality
  - the participants are immersed in a remote, synthetic world
  - a networked virtual environment (NVE)

Benford et al., 1998

Features of NVEs 1 (2)

- A shared sense of space
  - illusion of being located in the same place
  - same characteristics for all participants
    - time of day, weather, acoustics, haptics...

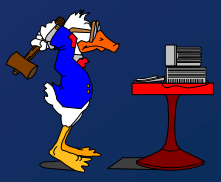


- A shared sense of presence
  - a participant has a virtual persona, an *avatar*
    - graphical presentation, body structure model, motion model, physical model, etc.
  - entering and leaving is visible for other participants
  - all participants do not have to be human-controlled

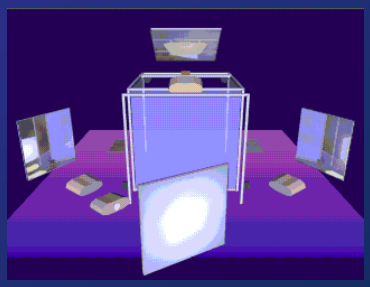
Features of NVEs 2 (2)

- A shared sense of time
  - see other participants' actions when they occur
    - enables real-time interaction

- A way to communicate
  - by gesture, by typed text, by voice...
- A way to share
  - interact realistically not only with each other but also with the virtual environment itself



Basic Components of (N)VE System 1 (2)



- Graphic engines and displays
  - the cornerstone of the NVE user interface
  - head-mounted displays (HMD)
  - Cave Automatic Virtual Environment (CAVE)
- Control and communication devices
  - keyboard, mouse
  - joystick
  - dataglove
  - HMD
  - motion detectors in full-body immersive environments
  - microphone

Basic Components of (N)VE System 2 (2)

- Processing systems
  - NVEs demand a considerable amount of processing capacity
  - computes the effects of the user's actions
  - determines when to notify other users
  - receives information from other users
  - controls autonomous objects
  - computes a visualization of the virtual environment
- Data network
  - exchange information
  - notify about environment changes
  - synchronize the shared state
  - communication among users

Challenges in Design and Development 1 (3)

- Difficult to implement correctly and effectively
- Include multiple traditional software types
- NVEs are
  - distributed systems
    - contend with managing network resources, data loss, network failure, concurrency
  - graphical applications
    - maintain real-time display frame rate
    - allocate the CPU among several tasks
  - interactive applications
    - process real-time input
    - users should see the virtual environment as if it exists locally

Challenges in Design and Development 2 (3)

- NVEs must work with other applications
  - typically integrate with database systems
  - need to support user authentication and may interact with commerce and other transaction systems
  - to support reproducible systems, must be able to log events in real-time to a persistent storage
    - the complete state of the NVE may not be known at any single host
- Optimizing one element of the NVE is hazardous
- Consider as a unified system
- NVE development is a difficult balancing act of trade-offs

Challenges in Design and Development 3 (3)

Balancing of

i. Network bandwidth
ii. Heterogeneity
iii. Distributed interaction
iv. Real-time system design and resource management
v. Failure management
vi. Scalability
vii. Deployment and configuration

i. Network Bandwidth

- Amount of desired information varies
- Amount of users varies
- How to allocate a limited network capacity?

ii. Heterogeneity

- Users do not have equipment with the same quality
- Whether to expose or hide the differences between participants
  - connection speed, processing capacity,…
- Hide by reducing the system to the lowest common denominator
  - a single ‘bad’ participant causes problems for everybody else
- Take a full advance of the available resources
  - user receive different levels of information
  - fair play?
- Graphical display, computational, and audio capabilities

iii. Distributed Interaction

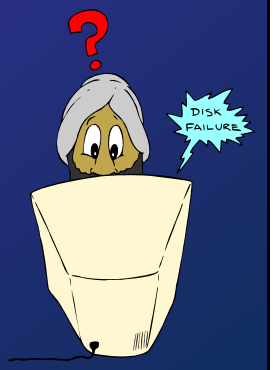
- One of the defining qualities of an NVE system
- NVE system must provide each user with the illusion that
  - the entire environment is located on the local machine
  - the actions of the users have a direct and immediate impact on the environment
- Difficult because of the messaging required
- Each host attempts to
  - present a consistent real-time view
  - cope with out-of-date information
- Problems when multiple users or components interact
  - collision detection, agreement, and resolution among participants

iv. Real-time System Design and Resource Management

- Real-time interaction defines the process and thread architecture
  - many tasks have hard real-time constraints
- Support quick detection and processing of user action
  - graphical image generation at fixed rate
  - network packets arrive asynchronously, process them soon
  - perform physics modelling and collision detection
- Everything in a single thread, use round-robin
- Segment into multiple threads, balance them
  - shared data structures on each host
  - shared locks

v. Failure Management

- One or more of the connected hosts can crash at any time
- Network connections can fail



Categories of failure handling:

1. System stop
   - entire NVE terminates due to a missing resource
2. System closure
   - no impact on the existing users but new ones are unable to login
3. System hindrance
   - a required service becomes unvaible; degrades the experience
4. System continuance
   - a non-critical service becomes unavailable; no noticeable effect

vi. Scalability

- Can be measured with the number of entities that may simultaneously participate in the system
  - may include human- and computer-controlled vehicles, a terrain, and even logical objects
- Also, the number of hosts, and physical distance between the hosts
- Depends on a variety of factors
  - network capacity, processor capabilities, rendering speeds,…
- The complexity of an NVE increases exponentially with the number of entities because of the number possible interactions between them
- Expensive to achieve because it requires enhancements to virtually all aspects of the NVE system

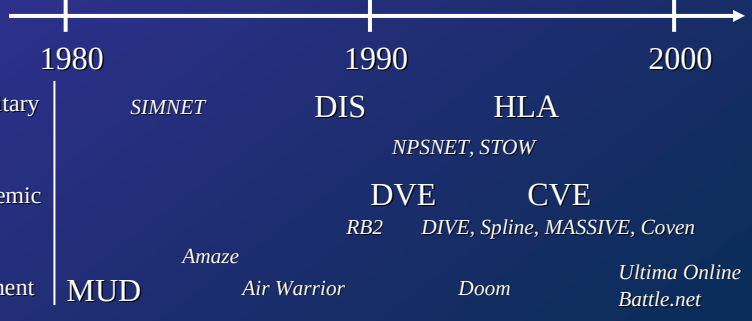
vii. Deployment and Configuration

- Deploying the software to participants
  - if the software is large, it is inappropriate for downloading
  - a small core library with dynamically downloaded components
- Implications to the software design, implementation language, and supported platforms
- In the case of web browsers or light-weight platforms, ensure that the enviroment
  - can be easily downloaded
  - conforms the security bounds
  - executes and displays correctly across different platforms
- Participants need an access to the configuration information
  - network addresses, encryption keys, access codes, images, computational modes,...

§2 Background

- Department of Defense (DoD)
  - SIMNET
  - Distributed Interactive Simulation (DIS)
  - High-Level Architecture (HLA)
- Academic NVEs
  - NPSNET
  - PARADISE
  - DIVE
  - BrickNet
  - other academic projects
- Networked games and demos
  - SGI *Flight* and *Dogfight*
  - *Doom*
  - other multiplayer games

History and Evolution



Network Software Architecture (NSA)

- NSA includes the inseparable issues of
  - what network protocol is used for the system and
  - what software architecture supports that protocol,

  within the confines of the available bandwidth and processor capacity.
- Important to solve *both* problems at once!

U.S. Department of Defense (DoD)

- The largest developer of NVEs for use as simulation systems
  - one of the first to develop NVEs with its SIMNET system
  - the first to do work on large-scale NVEs
- SIMNET (simulator networking)
  - begun 1983, delivered 1990
  - a distributed military virtual environment developed for DARPA (Defense Advanced Research Projects Agency)
  - develop a 'low-cost' NVE for training small units (tanks, helicopters,…) to fight as a team