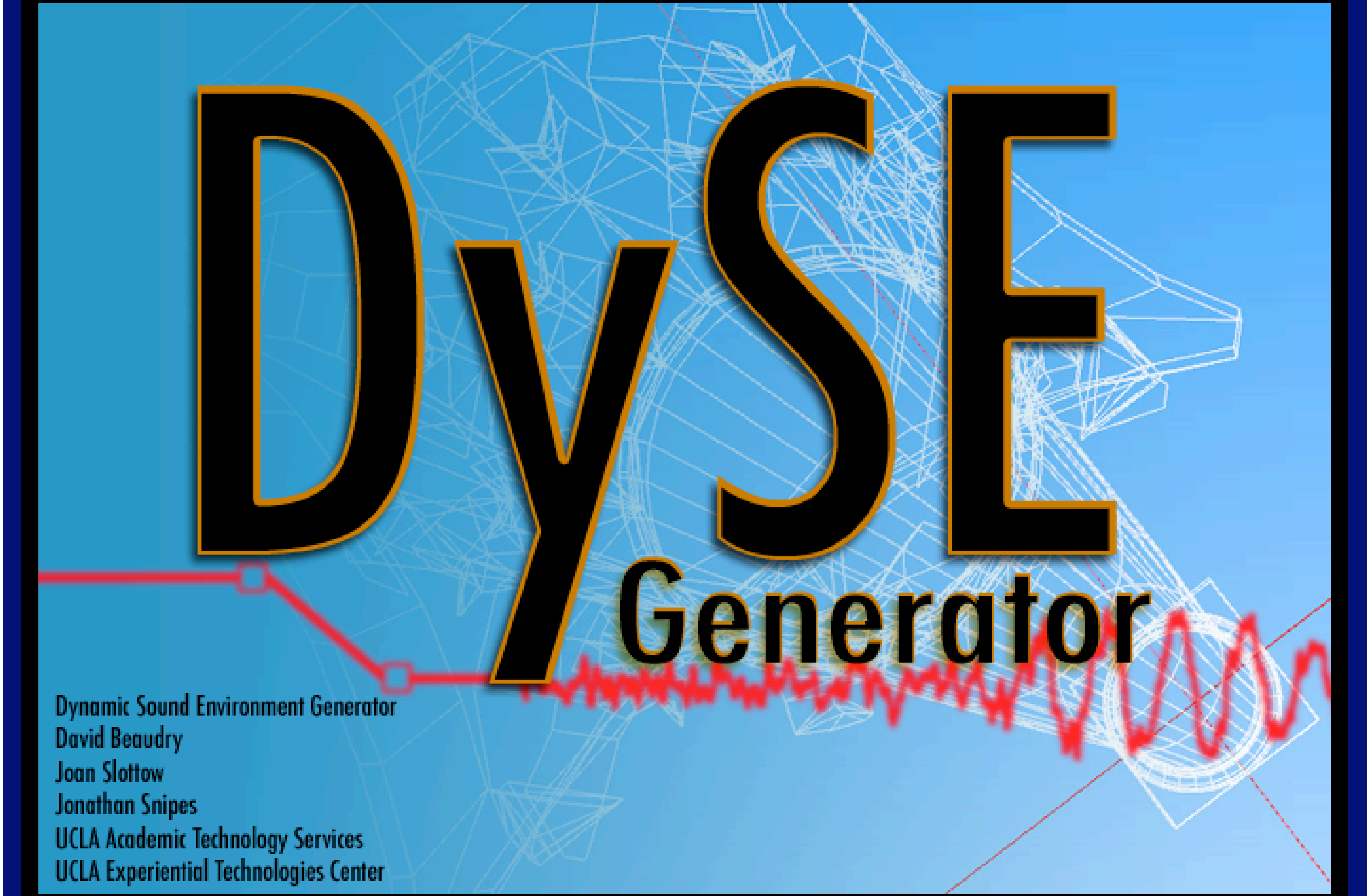
Poster Session:
Gallery of Projects Enabled by OSC

Session Chair: Matthew Wright

DySE Generator: A sound design tool for virtual reality applications

David Beaudry, Virtual Reality
Audio Specialist, UCLA
Visualization Portal

306 Soda Hall (This room)



DySE Generator

Dynamic Sound Environment Generator

David Beaudry
Joan Slottow
Jonathan Snipes
UCLA Academic Technology Services
UCLA Experiential Technologies Center

SonART
A new multimedia environment for networked collaboration

Woon Yeo, PhD Candidate,

Stanford University Center for Computer Research in Music and Acoustics (CCRMA)

310 Soda Hall

Human/Computer Interaction projects at CCRMA

Michael Gurevich, PhD Candidate,

Stanford University Center for Computer Research in Music and Acoustics (CCRMA)

320 Soda Hall

Quintet.Net:
An interactive performance environment for the Internet

Prof. Georg Hajdu
Master's Program in Multimedia and
Music, HfMT Hamburg

306 Soda Hall (This room)

Real-time Distributed Media Applications in LANs with OSC

Tristan Jehan, Dan Overholt,
Hugo Solís Garcia and Cati Vaucelle,
MIT Media Lab

306 Soda Hall (this room)

Max/MSP Programming Practice with OSC

David Wessel, Director, UC Berkeley
Center for New Music and Audio
Technologies (CNMAT)

320 Soda Hall

An OSC Driver Framework for Gesture Sensors

Stephen Pope,
UC Santa Barbara Center for
Research in Electronic Art
Technology (CREATE)

306 Soda Hall (this room)

Building Large-scale Interactive Systems with OSC, Siren, CSL, and CRAM

Stephen Travis Pope

Center for Research in Electronic Art Technology (CREATE)
Graduate Program in Media Arts and Technologies (MAT)
University of California, Santa Barbara (UCSB)
stp@{create,mat}.ucsb.edu

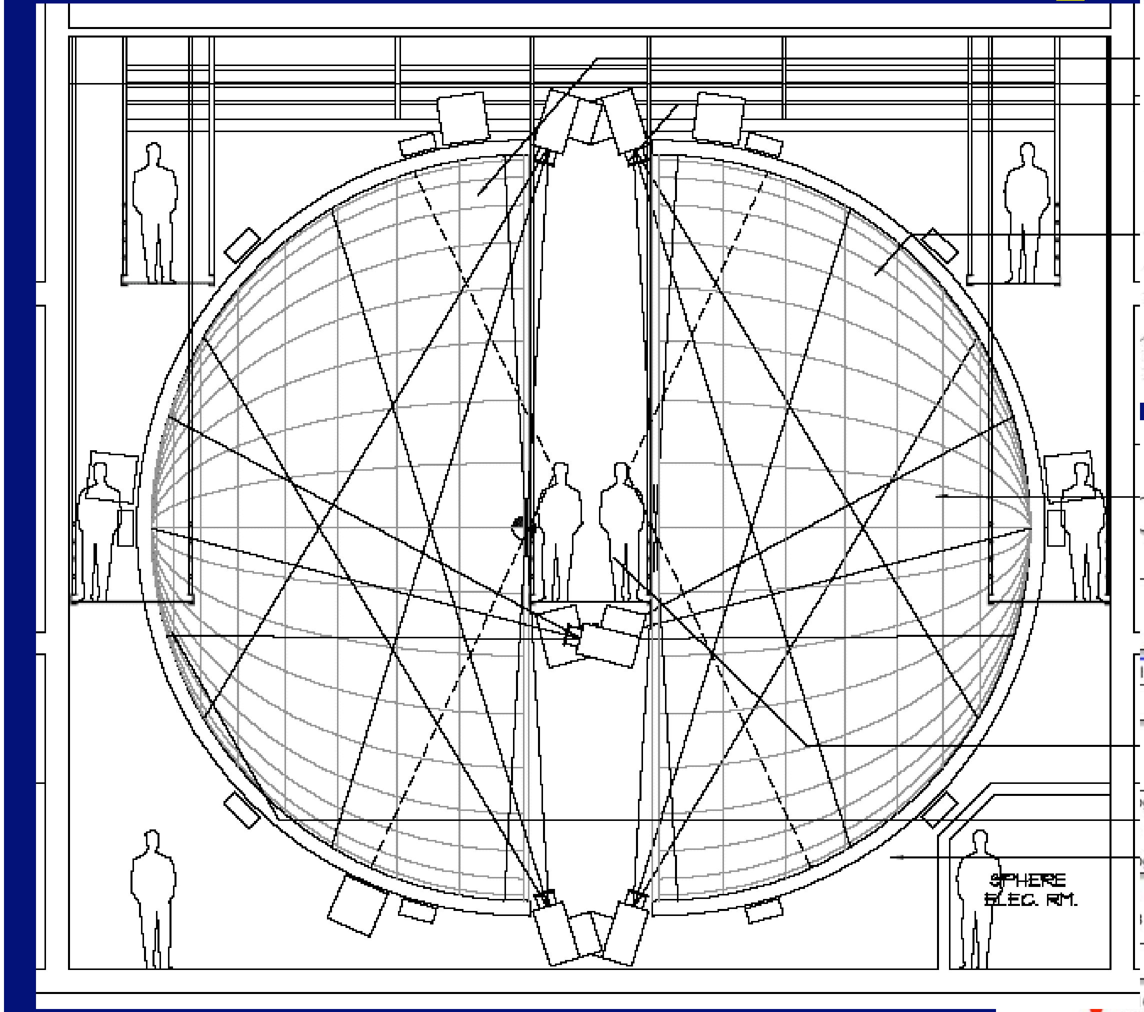
HW/SW Components

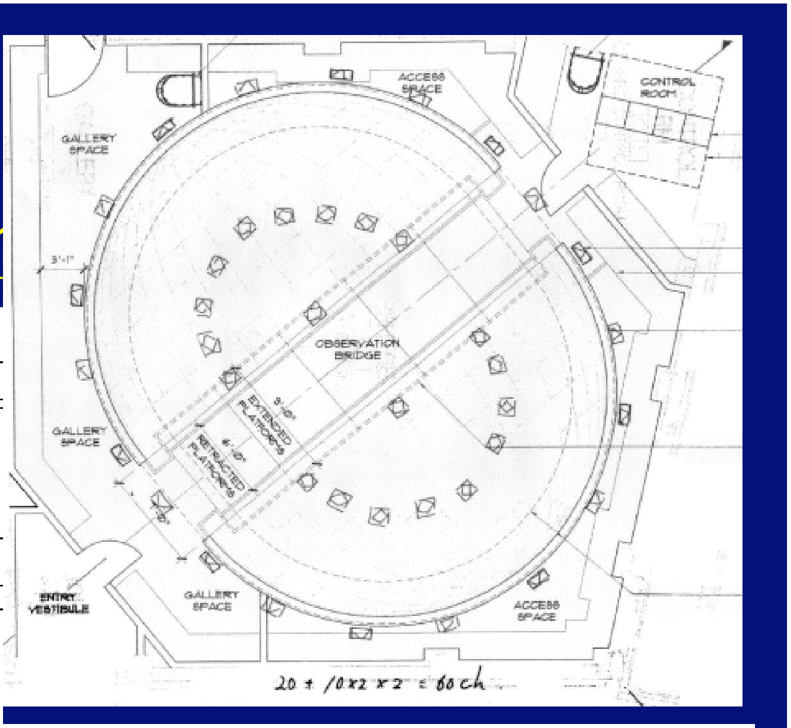
- **Siren**: Hierarchical/procedural representation for composers (OSC out)
- **CSL**: Scalable DSP framework (OSC srv)
- **CRAM**: Cluster management for distributed RT OO software (Mgr)
- **CNSI Sphere**: A really cool loud/ bright/sensing space to play in!

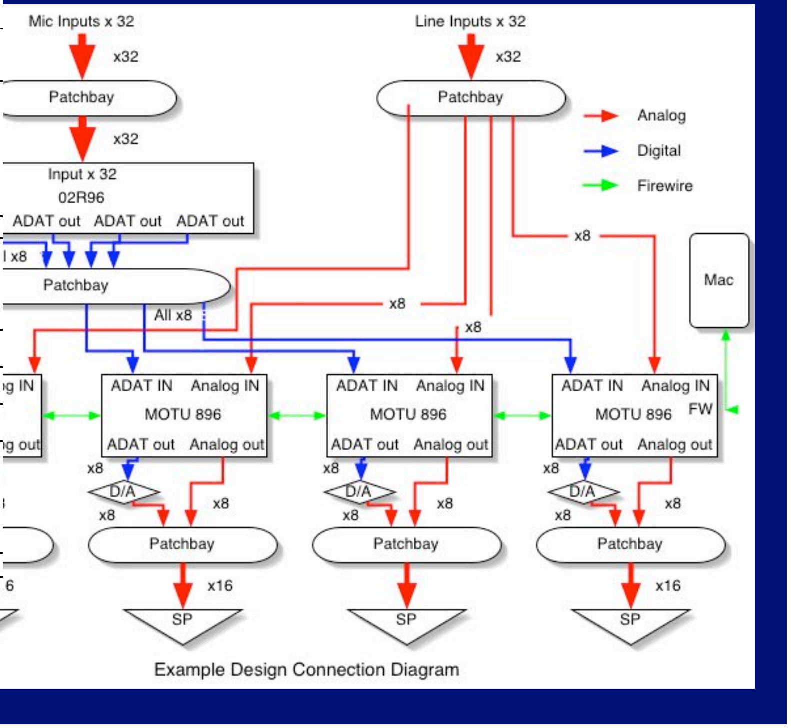
Cal. NanoSystems Inst. @ UCSB

- MAT in CNSI: labs, studios, workshops, sphere
- CNSI compute infrastructure
  - Traditional vector supercomputer
  - 1024-node Linux cluster
  - Multimedia processing cluster (TBD)
- Sphere: 3-story I/O space
  - 12-channel overlapping video output
  - 128-channel sound output
  - Camera/microphone/sensor multi-modal input

CNSI Sph

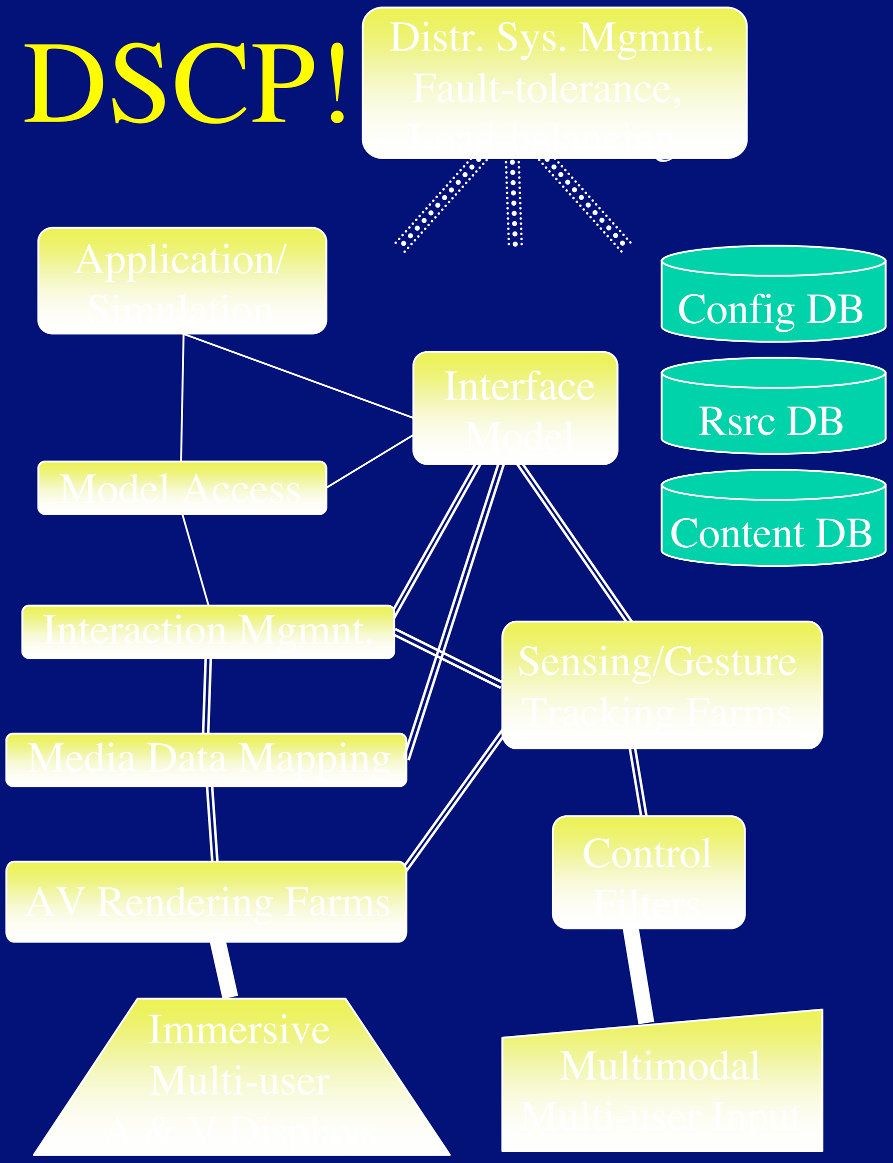






Example Design Connection Diagram

How? DSCP!



<u>Distributed Sensing, Computation, and Projection</u> = MVC on steroids

Back-end application models are scientific/numerical/simulation

Multimodal multiuser **sensing/control** and tracking/mapping farms

Application = sensing/tracking policies + output data mappings

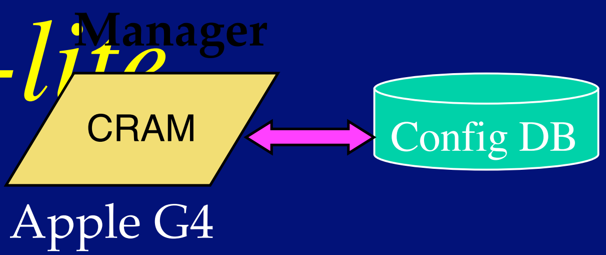
Presentation/interaction via CNSI Sphere, LAN/WAN streaming

Infrastructure uses CRAM mgmnt

DBs for configurations, resources, and media content (renderers)

]

Current *Sphere-lite*

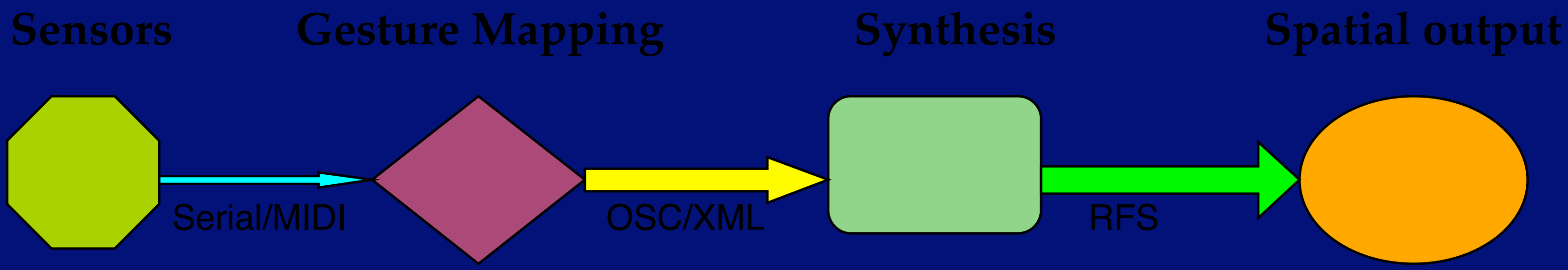


Manager

CRAM

Config DB

Apple G4



Sensors

Gesture Mapping

Synthesis

Spatial output

Serial/MIDI

OSC/XML

RFS

MIDI

Matrix

Graphonic

OT_Kbd

Creatovox

VR Trackers

Mot. Capture

AdC_Panner

(LAN is switched 1000BaseT)

Wintel PCs

Apple G5s

Apple Xserves

Apple G5s/G4s

Echo Layla

Sun Ultras

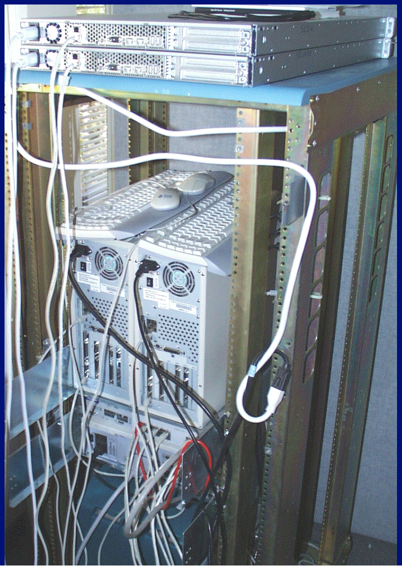
SGI Octanes

Interfaces

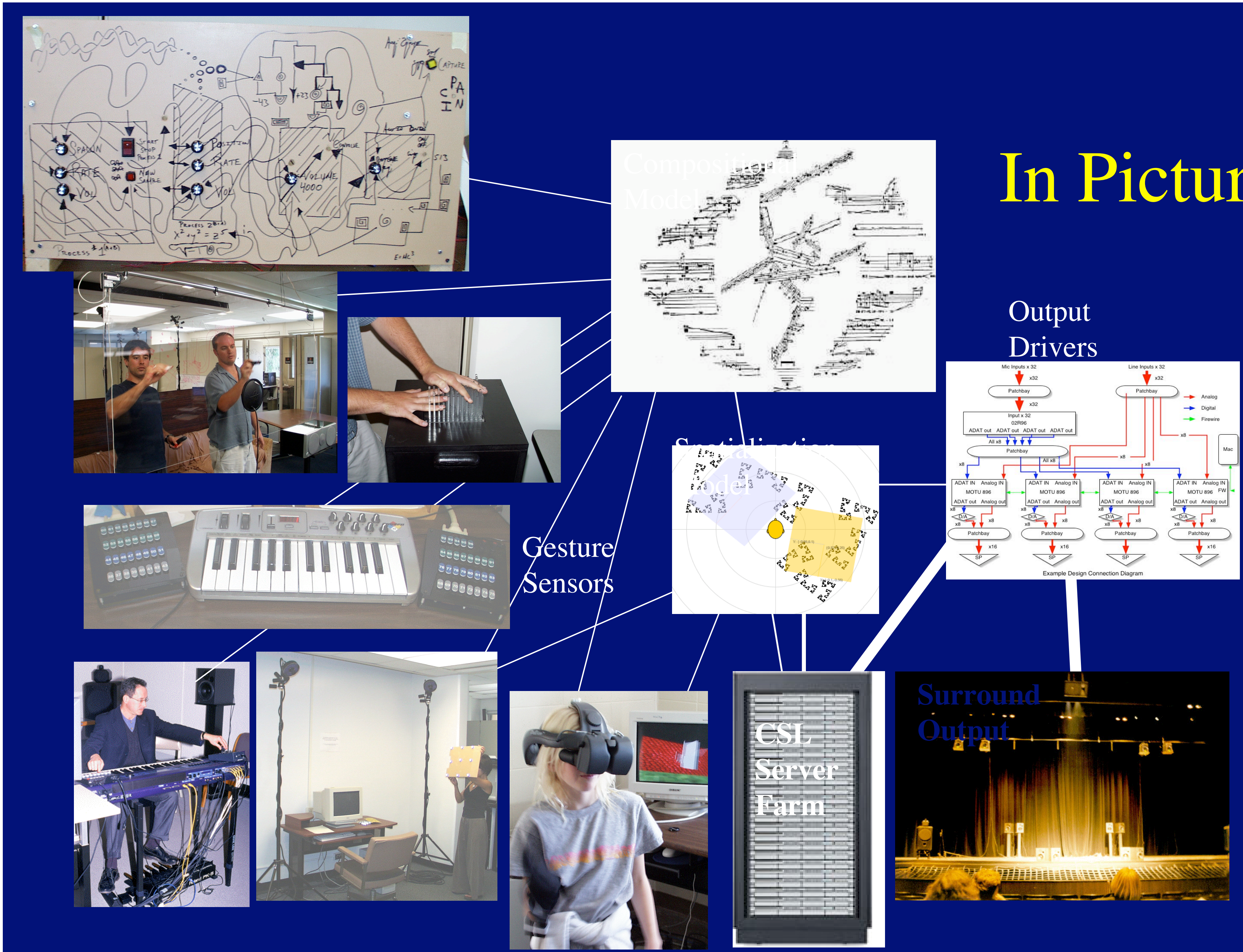
Occam/Macco

OscAR

Apple G5



In Pictures



Compositional Model

Spatialization Model

Gesture Sensors

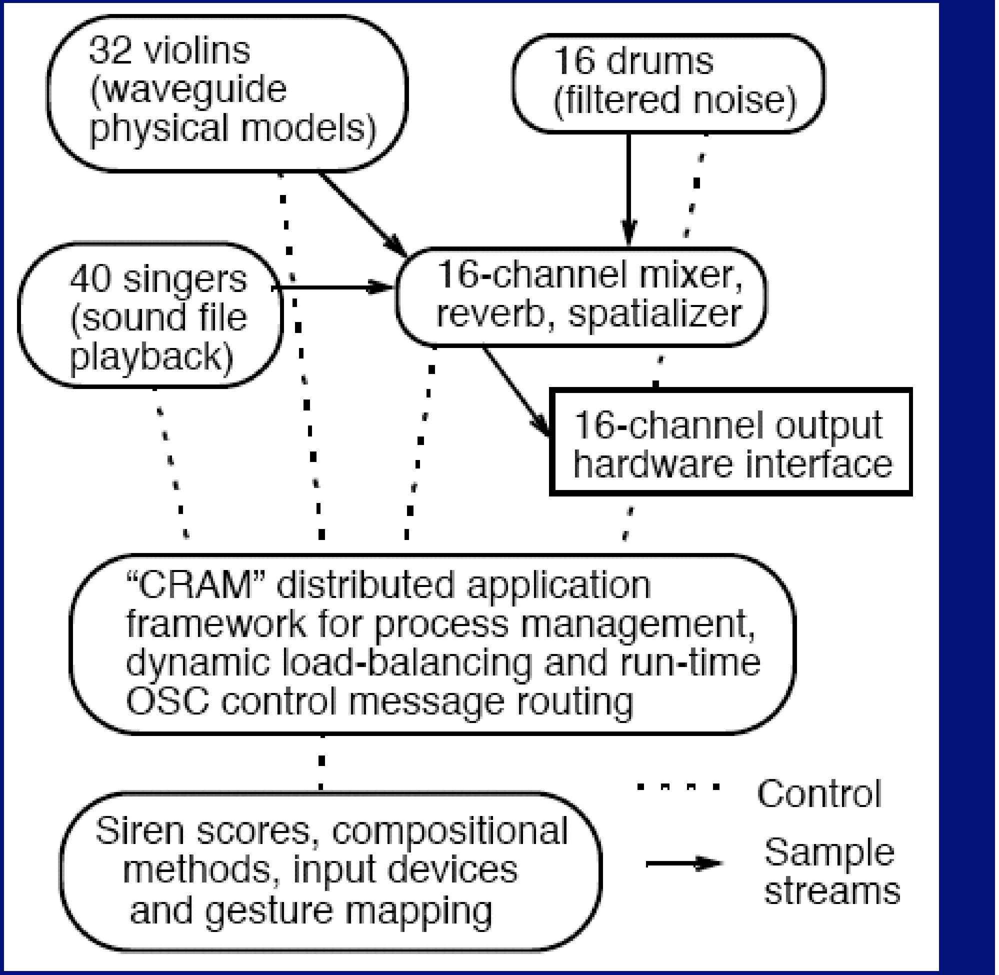
Output Drivers

CSL Server Farm

Surround Output

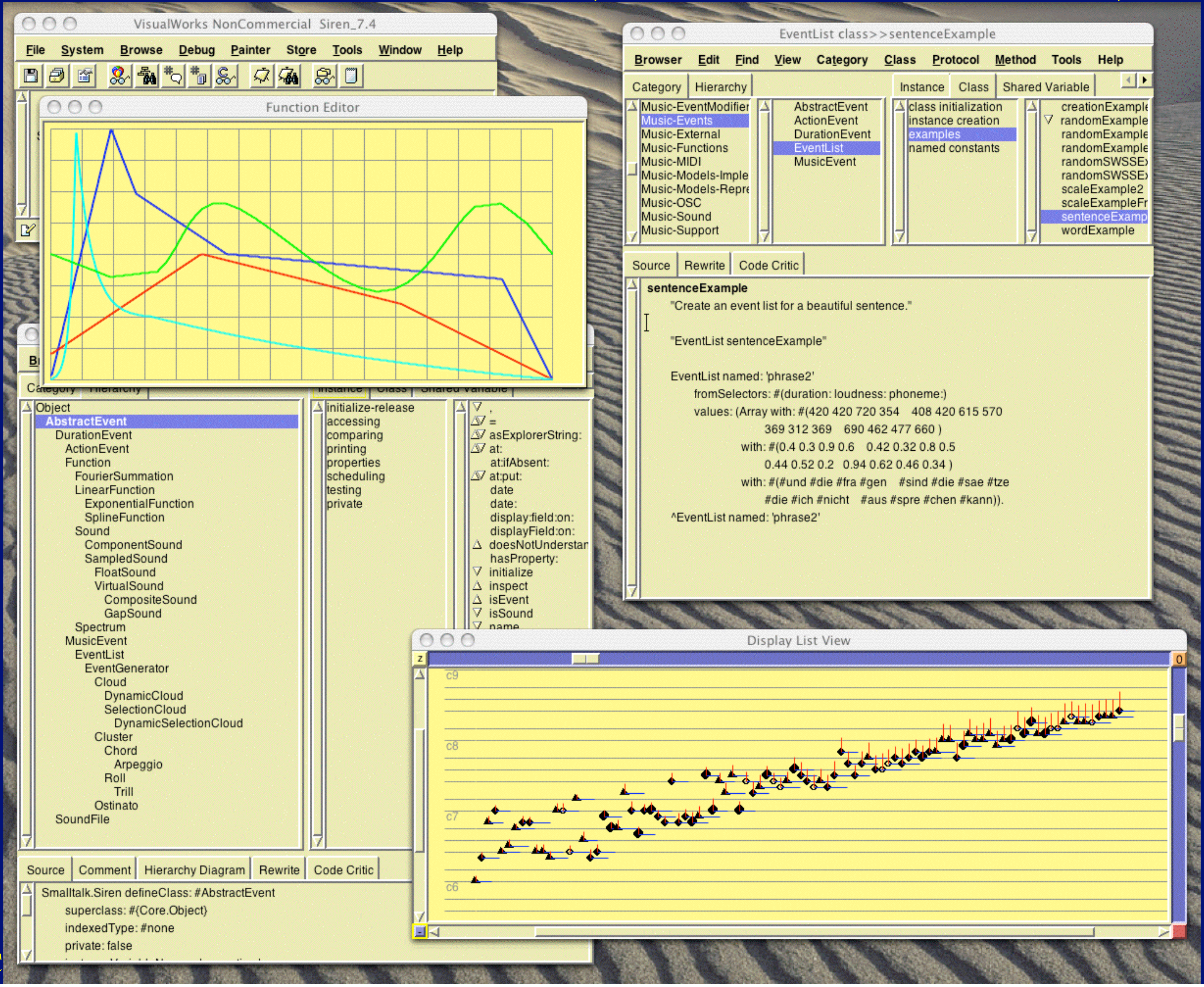
Networked Synthesis/Performance

- Managed “orchestra-scale” sound synthesis, multi-modal gestural sensing and control, and pluriphonic projection (up to 128 channel output in the CNSI sphere)


32 violins (waveguide physical models)
16 drums (filtered noise)
40 singers (sound file playback)
16-channel mixer, reverb, spatializer
16-channel output hardware interface
“CRAM” distributed application framework for process management, dynamic load-balancing and run-time OSC control message routing
Siren scores, compositional methods, input devices and gesture mapping
Control
Sample streams

Siren 2003 (VisualWorks)

Demo



```
sentenceExample
	"Create an event list for a beautiful sentence."

	"EventList sentenceExample"

	EventList named: 'phrase2'
		fromSelectors: #(duration: loudness: phoneme:)
		values: (Array with: #(420 420 720 354    408 420 615 570
					369 312 369    690 462 477 660 )
				with: #(0.4 0.3 0.9 0.6    0.42 0.32 0.8 0.5
					0.44 0.52 0.2   0.94 0.62 0.46 0.34 )
				with: #(#und #die #fra #gen    #sind #die #sae #tze
					#die #ich #nicht    #aus #spre #chen #kann)).
	^EventList named: 'phrase2'
```

```
Smalltalk.Siren defineClass: #AbstractEvent
	superclass: #{Core.Object}
	indexedType: #none
	private: false
```

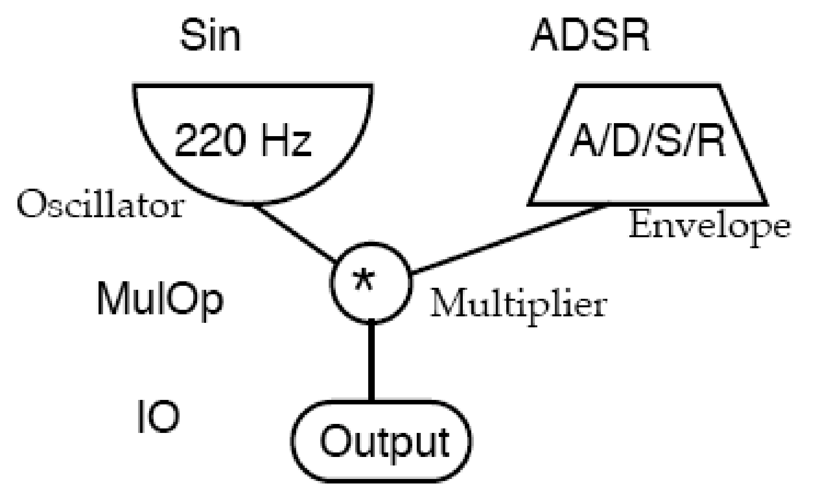
Open Sound C

CSL "Hello world" Program

Demo

Sine wave with envelope

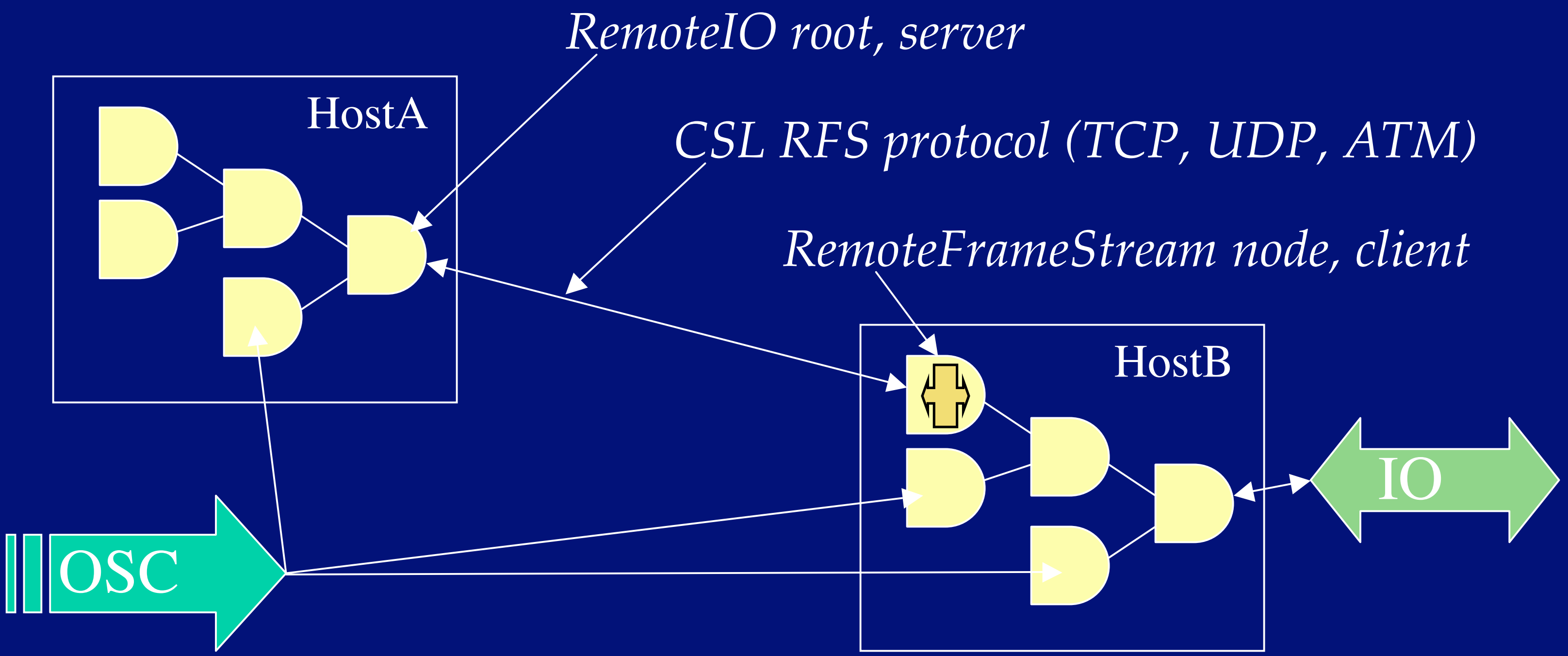
```
// Create a sine oscillator -- this is a comment

        Sine osc(220.0);

// Create an ADSR envelope -- args are (dur, a, d, s, r)

        ADSR env(3.0, 0.06, 0.2, 0.2, 1.5);

// Create a multiplier

        MulOp mul(osc, env);

// Plug it into the output driver

        globalIO.set_root(mul);
```



Multi-host CSL Graphs

Demo

- Distributed sub-graph processing with RemoteIO and RemoteFrameStream, RFS protocol, buffering



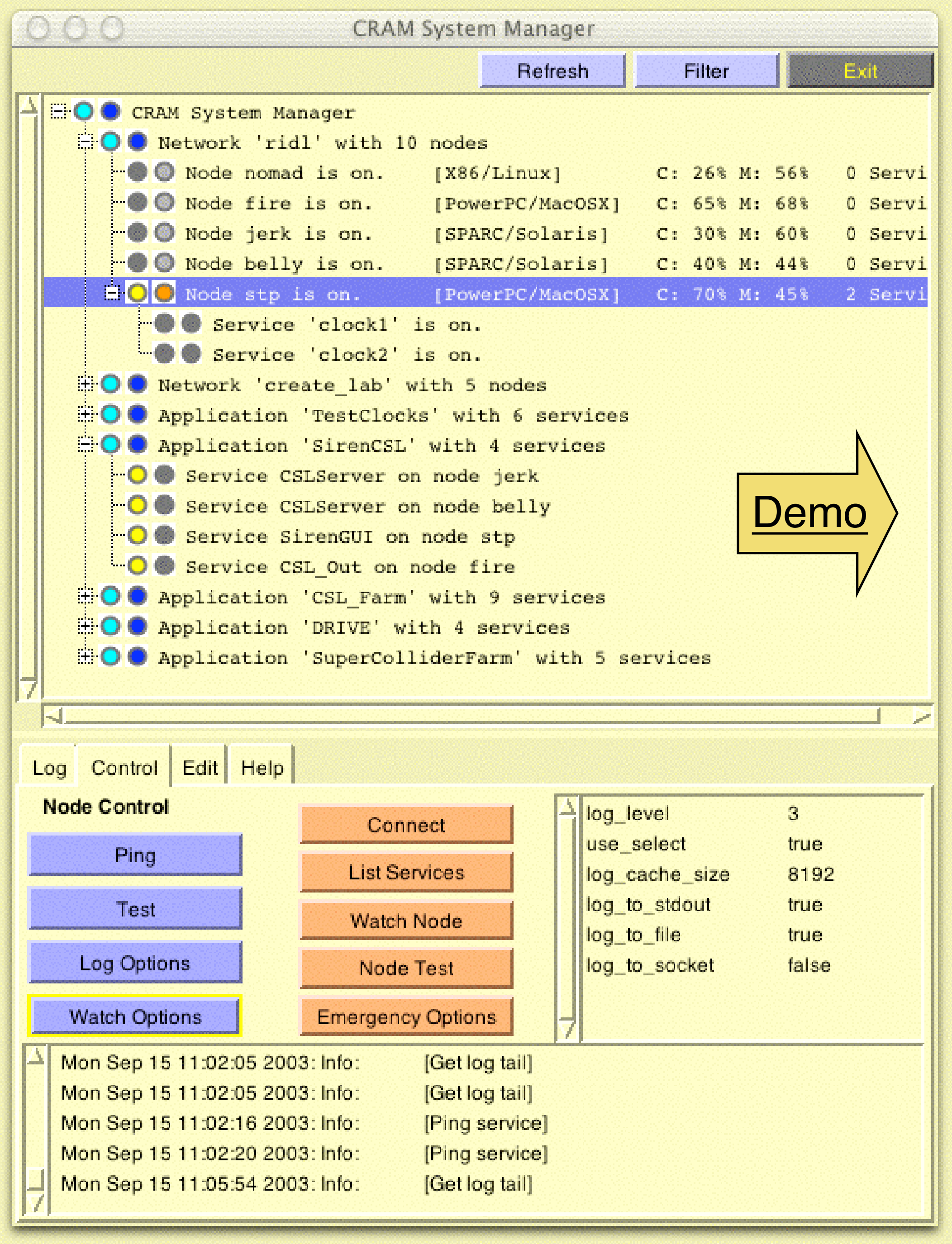
RemoteIO root, server

CSL RFS protocol (TCP, UDP, ATM)

RemoteFrameStream node, client

CRAM Manager

- Network/Node
- Node/Service
- Application/Service
- Log/Control pane
  - Run-time monitor
  - Planning
  - DB play-back



```
CRAM System Manager
  Network 'ridl' with 10 nodes
    Node nomad is on.     [X86/Linux]        C: 26% M: 56%   0 Servi
    Node fire is on.      [PowerPC/MacOSX]   C: 65% M: 68%   0 Servi
    Node jerk is on.      [SPARC/Solaris]    C: 30% M: 60%   0 Servi
    Node belly is on.     [SPARC/Solaris]    C: 40% M: 44%   0 Servi
    Node stp is on.       [PowerPC/MacOSX]   C: 70% M: 45%   2 Servi
      Service 'clock1' is on.
      Service 'clock2' is on.
  Network 'create_lab' with 5 nodes
  Application 'TestClocks' with 6 services
  Application 'SirenCSL' with 4 services
    Service CSLServer on node jerk
    Service CSLServer on node belly
    Service SirenGUI on node stp
    Service CSL_Out on node fire
  Application 'CSL_Farm' with 9 services
  Application 'DRIVE' with 4 services
  Application 'SuperColliderFarm' with 5 services
```

```
log_level         3
use_select        true
log_cache_size    8192
log_to_stdout     true
log_to_file       true
log_to_socket     false
```

```
Mon Sep 15 11:02:05 2003: Info:     [Get log tail]
Mon Sep 15 11:02:05 2003: Info:     [Get log tail]
Mon Sep 15 11:02:16 2003: Info:     [Ping service]
Mon Sep 15 11:02:20 2003: Info:     [Ping service]
Mon Sep 15 11:05:54 2003: Info:     [Get log tail]
```

GestureSensor Drivers & Servers

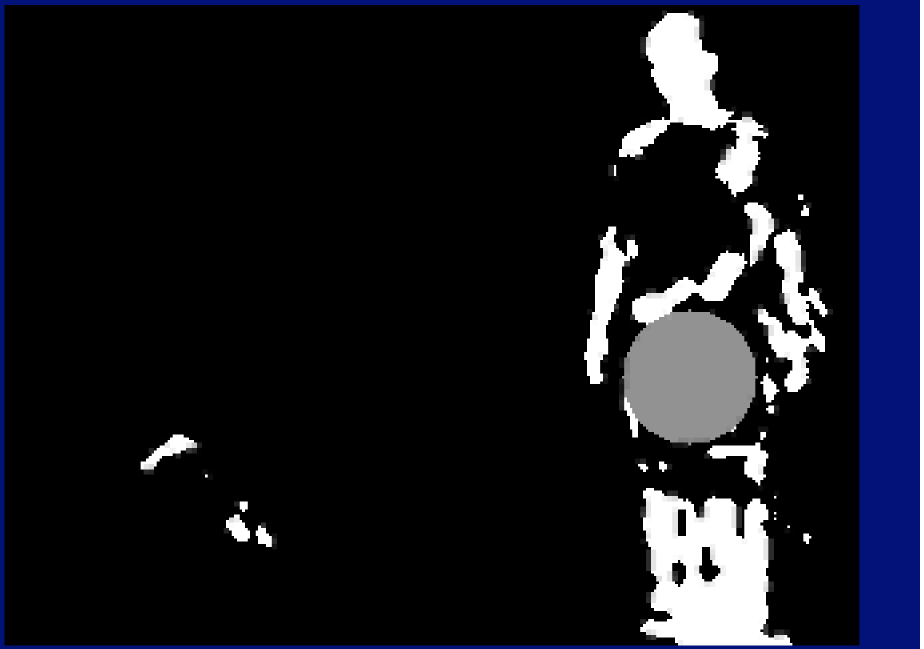
- Reusable sensor driver framework
  - Serial in, cacheing/differencing/throttling, OSC out
- GestureSensors: receive OSC or MIDI

```
void * mData;          // data array (typically a float *)
char * mCmd;           // OSC command (without the '/')
char * mTypeString;    // OSC type string, e.g., "ffff"
```

  - Event input thread mgmnt
  - Parsing and differencing
  - Map to static or global data or messages
- Subclasses
  - Glove, Ebeam, Matrix, FOBirds, AdC_Panner, etc.

CV-to-OSC

- **Multiple-camera 3D motion tracking of multiple sources**
- **Data mapping for sound synthesis and transformation algorithms**
- **Intelligent trans-media system that learns and adapts, based on memory of the actions and states of the sensor space**



OSC Control of VST Plug-ins

Michael Zbyszynski and Adrian Freed,
UC Berkeley Center for New Music
and Audio Technologies (CNMAT)

310 Soda Hall

Go to the posters!

306 Soda (this room):

- Beaudry & Slottow: DySE Generator: A sound design tool for virtual reality applications
- Hajdu: Quintet.Net: An interactive perf. env. for the Internet
- Pope: An OSC Driver Framework for Gesture Sensors
- Jehan et al: Real-time Dist. Media Apps. in LANs with OSC

310 Soda

- Zbyszynski & Freed: OSC Control of VST Plug-ins
- Yeo: SonART: A new multimedia environment for networked collaboration

320 Soda

- Wessel: Max/MSP Programming Practice with OSC
- Gurevich: HCI projects at CCRMA