

The Interim DynaPiano



The *Interim DynaPiano*: An Integrated Computer Tool and Instrument for Composers

Stephen Travis Pope

The Nomad Group, *Computer Music Journal*, CCRMA
P. O. Box 60632, Palo Alto, California 94306 USA
stp@CCRMA.Stanford.edu

Motivation



Why is he doing this?

- Background
 - SW Sound Synthesis
 - SSSP
 - Cadmus 9230/m
 - HyperScore ToolKit and MODE
 - The need for a Tool *and* Instrument
 - The need for (trans)portability

Introduction



Outline

Background
IDP Hardware Configuration
MODE Software
IDP/MODE Application Examples
Conclusions, Directions, Availability

- The IDP System: a Hardware/Software tool for composers
- The Name and its History
- Applications—Composition, Realization and Performance

I W-B CM S



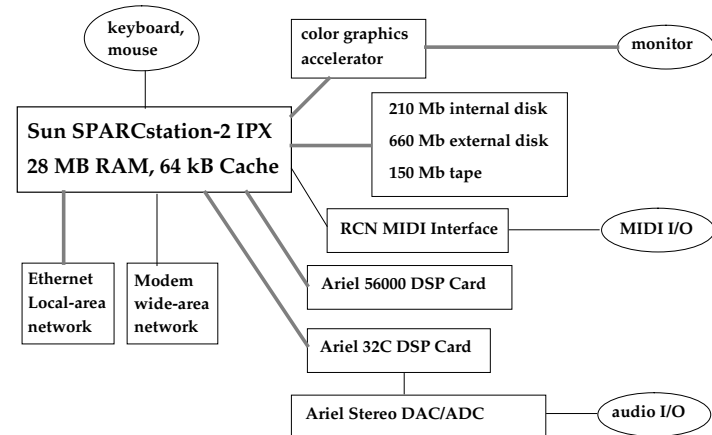
- Ten Years of Integrated workstation-based computer music systems using relatively-stable base technology
- Components (same as in 1980)
 - Commercial engineering workstation
 - Large RAM and disk memories
 - Multi-tasking OS
 - Real-time sound and MIDI I/O
 - C-based music and DSP libraries
 - Interactive SW development/delivery environment (Lisp or Smalltalk)

I W-B CM S Examples

- **Examples**
 - PDP-1 1/SSSP (?)
 - CHANT/FORMES
 - Cadmus 9230/m + CARL
 - LM-2 + Flavors Band
 - Mac + Kyma/Capybara
 - NeXT + Common Lisp Music
 - IRCAM Musical Workstation
- **Non-Examples**
 - Mac + MIDI
 - NeXT + Objective C
 - DSP/Hard disk recorders

IDP Hardware

Current IDP Hardware Configuration

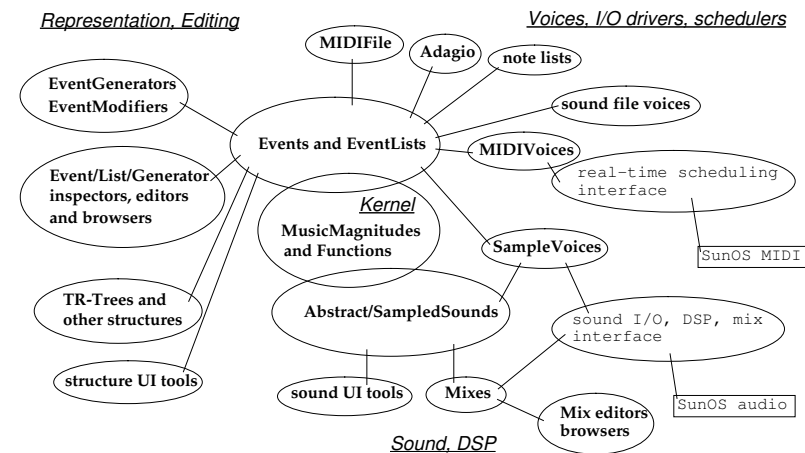


The MODE Software

A Smalltalk-80-based framework and tool kit
for music composition and performance

- **SmOke music representation**
 - Music Magnitudes
 - Events, Event Lists
 - Generators, Modifiers, and Structures
 - Functions, Sounds and DSP
- **MODE voices and I/O**
 - Voices as device drivers
 - Voices as property-to-parameter mappers
- **MODE user interface components**
 - MVC Framework and Navigator
 - MODE MVC Support and Applications

MODE Software Components



SmOke Music Magnitudes and Events



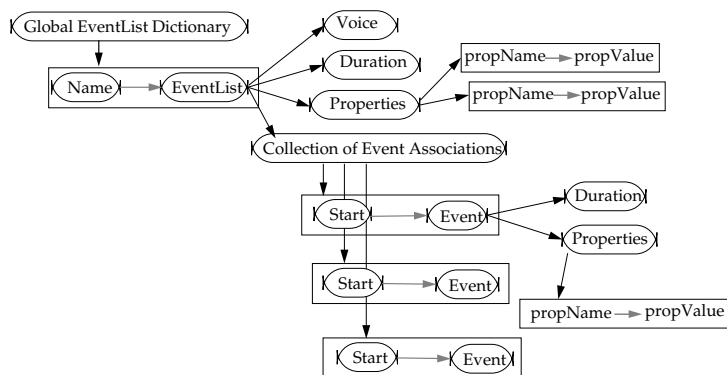
Music magnitude models and concrete classes

- **Pitch, Amplitude, Duration models**
- **HertzPitch** [440.0 Hz], **SymbolicPitch** ['f#2' pitch]
BeatDuration [1/4 beat], **SymbolicLoudness** ['mp' ampl]
- [(1/4 beat) + 80 msec] ['e4' pitch + 12 cents + 4 Hz]

Abstract events and event lists

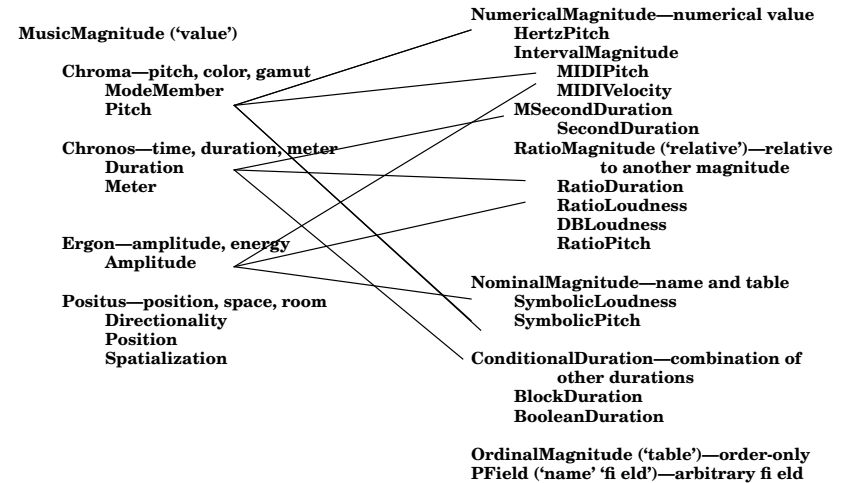
- **Event as property list**
- **Event properties can look behavioral**
 (declarative) [evt *voice*: 'flute'] or static
 (procedural) [evt *at*: #voice *put*: 'flute']
- **Voices or applications determine the semantics of properties using coercion**
 (e.g., *anEvent pitch asMIDI, asHz, ...*)

The SmOke Event List Structure



Example Event and Event List State

Music Magnitude Species and Classes



Verbose and Terse SmOke Examples



```

"Verbose MusicMagnitude Creation
and Coercion Messages"
"Answers Duration 62 msec."
(Duration value: 1/16) asMsec
"Answers Pitch 261.623 Hz."
(Pitch value: 60) asHertz
"Answers Amplitude 106."
(Amplitude value: 'ff') asMIDI

```

```

"Event Creation Messages"
"Create a 'generic' event."
Event dur: 1/4 pitch: 'c3' ampl: 'mf'
"Create one with added props."
(Event dur: 1/4 pitch: 'c3')
color: #green; accent: #sfz

"EventList Usage"
"Create a named event list."
el := EventList newNamed: #demo1.
"Add an event to it at time 0."
el add: (Event dur: 1 pitch: 36 ampl: 'mf');
"Add an event after the first."
add: (Event dur: 1 pitch: 40 ampl: 'mf');

"Add a sublist with 3 events"
add: (EventList new"a chord."
add: (Event dur: 1 pitch: 36) at: 0;
add: (Event dur: 1 pitch: 40) at: 0;
add: (Event dur: 1 pitch: 43) at: 0)

```

Verbose SmOke Examples

```

"Terse MusicMagnitude Creation using post-ops"
440 Hz 250 msec
1/4 beat 'c#3' pitch

```

```

"Terse Events using concat. of music mags"
440 Hz, 1/4 beat, -12 dB, (#voice -> #flute).
38 key, 280 ticks, 56 vel.
(#c4 pitch, 0.21 sec, 0.37 ampl).

```

```

"Terse EventLists using concat. of events or
(duration -> event) associations"
(440 Hz, (1/1 beat), 44.7 dB, "comma"
(1 => ((1.396 sec, 0.714 ampl) word: #xu))
"Each Example—First measure of Fugue 2 from WTK
(ignoring the initial rest)."

```

```

(((0 beat) => (1/16 beat, 'c3' pitch)),
((1/16 beat) => ('b2' pitch)),
((1/8 beat) => (1/8 beat, 'c3' pitch))),
((1/4 beat) => ('g2' pitch)),
((3/8 beat) => ('a-flat2' pitch)),
((1/2 beat) => ((1/16 beat, 'c3' pitch),
((1/16 beat) => ('b2' pitch)),
((1/8 beat) => (1/8 beat, 'c3' pitch))),
((3/4 beat) => ('d3' pitch)),
((7/8 beat) => ('g2' pitch))

```

Terse SmOke Examples

MODE Event List Usage



Event List Creation

- Text, graphical input
- Procedural generation
- Reading data from other applications
- Built-in creation methods

EventList fromSelectors:data:

EventGenerators

EventGenerators and EventModifi ers

(see also [ICMC 1989])

TR-Trees and other structures

(see also [ICMC 1991])

Persistency, links and hypermedia

(HyperScore idea and tools)

Sampled Sound Support



• MODE Functions

Abstract Function (stored data or breakpoints)

Linear, Exponential, Spline, Fourier, . . .

• Sound, SampledSound, FloatSound, etc

Sound formats and Operations

• SoundFile

Headers and IO formats

Internal vs. external header parsing and generation

• SoundPort

Three ways to do it

Voices and Event I/O



Ways of Playing Event Lists

[eventList *playOn*: voice] or

[voice *play*: eventList *at*: time] or

[eventList *voice*: voice; *play*]

Voices as Property-to-Parameter Drivers

Examples:

- (NoteListVoice on: fi le play: list.
- (AdagioVoice on: fi le) eventList.
- (SoundVoice on: port) play: list.
- (MIDIVoice on: (aMIDIDevice on: aMIDIPort)) play: list.

MODE System Interface



Three versions of Sampled Sound and MIDI I/O

• Voices, Devices, and Ports

- UnixProcess interface to *splay*, *aplay*, *mplay*, *mread* (optionally *from/tosnd*)

• UserPrimitives

Not UNIX-specific, run-time UPs available

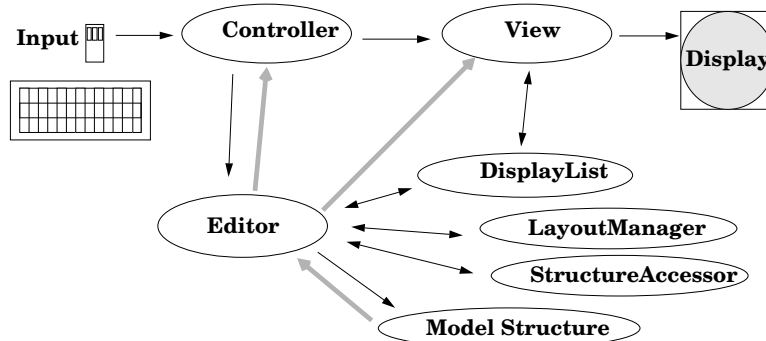
• Sockets and RemoteDoItServers—ST80

process spawns C-programs that read or write data from/to a socket

MODE User Interfaces



Model/View/Controller Programming and Navigator



MODE Sampled Sound UI



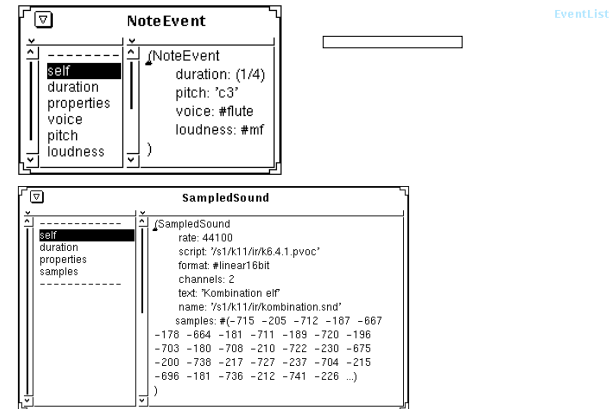
Sound Browser on snd

SoundBrowser

MODE UI Examples



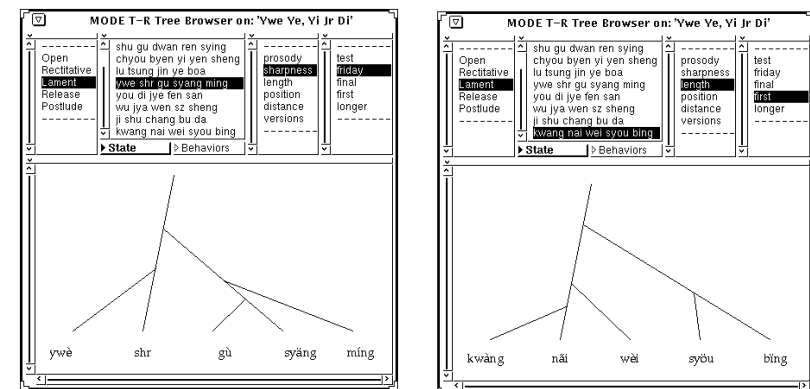
Event, EventList and Sound Inspectors (old)



MODE Structure Editors



TRTreeBrowsers (old)



LPC Vocoding Tools

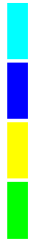


LPC Frame Data from: ./mode/data/ab.lpc

Function Editors



Function Editor



n Editor



MODE Menu and Play List



File List on mode/data/*

MODE

Ex view
Ex menu
UI Ex menu
Play List
Event Lists
Functions
Sounds
Mixer
LPC view
PVoc view
all notes off
all sound off

EventListEditors



Pitch-Time Editor

-Steffens Event List View

Mixer View



Mix View on: M2.d4b.sc

MODE Demo



Mode Demo Tools

- Example List View
- MODE Kernel Example Menu
- MODE UI Example Menu

- SmOke Representation
- Voices and I/O
- UI Examples and Tools

Conclusions, Directions



- IDP as an CMW , ICE and Instrument
- Porting IDP/MODE to new Platforms
HW: SS-10, NeXT, Indigo
L-L SW: Multi-UI, Sockets, ISIS, pmix
- Status—Hardware and software described here works with few exceptions(6/1992)
- A availability—MODE software, doc, etc. is available via anonymous Internet ftp from *ccrma-ftp.Stanford.edu* directory *pub/st80*
- Usage—Sound processing and live performance (as a sampler) in *Celebration*
- The Future

