The Siren Music/Sound Framework for Smalltalk

Demonstration Video

Stephen Travis Pope, May 2007

Siren Demo Outline

- Overview
- Smalltalk and Siren Set-up
- Smoke Music Representation
- Graphics and GUI Frameworks
- I/O and External Interfaces
- Advanced Applications and Editors
- Composition Examples

Part 1

- Smalltalk and Siren Set-up
  - Starting Smalltalk, Building Siren
  - Siren Utility Applications
- The Smoke Representation
  - Magnitudes, Events and Event Lists
  - EventGenerators, EventModifiers
  - Voices, Scheduling, Performance
  - Functions, Sounds, and Spectra

Part 2

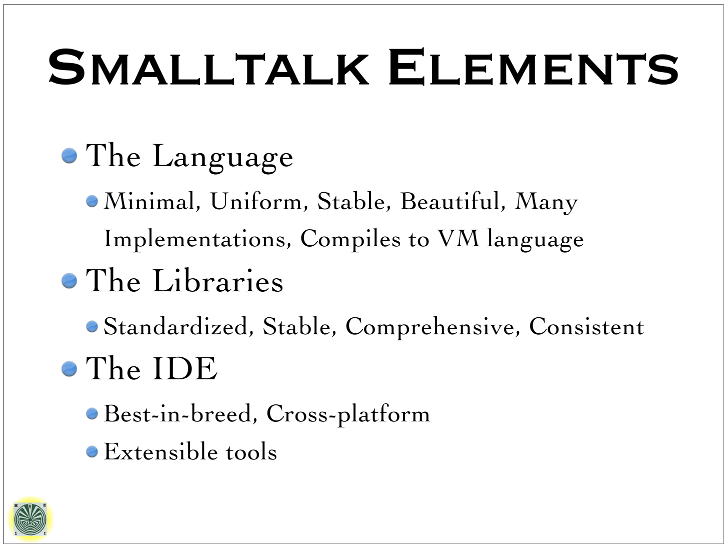
- Siren Graphics, GUI Frameworks
  - DisplayList/DisplayListView Framework
  - Siren Views, Controllers, Inspectors, Editors
- I/O and External Interfaces
  - Setting-up the DLLCC External Interfaces
  - File, MIDI, OpenSoundControl, Sound I/O
  - Interfaces to CSL, Loris and LPC

Part 3

- Advanced Applications, Editors
  - Smalltalk Tool Extensions
  - Loris, CSL, LPC Interfaces
  - EventList Editors
- Composition Examples
- Evaluation, Summary

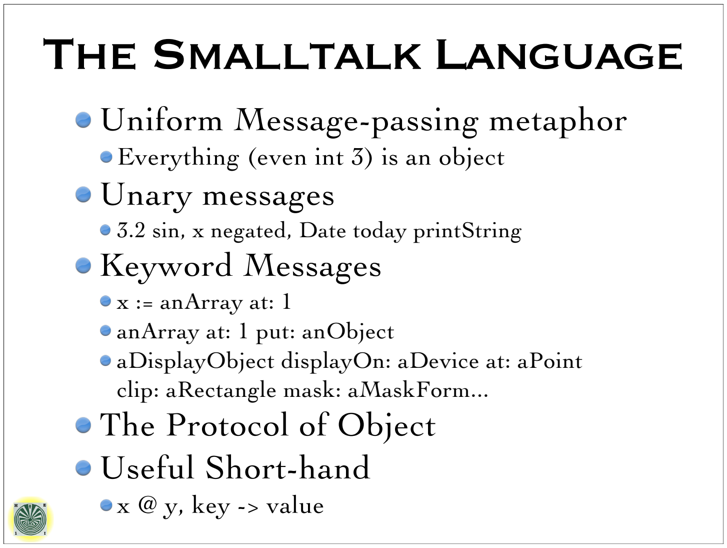
A Brief Tour of Smalltalk

- Smalltalk Programming System
  - Language, Libraries, Tools, Methodology
  - Smalltalk History (72, 76, 80, 80v2, ANSI)
  - Delivery System: VM and VI, srcs/changes
  - Current Implementations:
    - VisualWorks
    - Squeak
    - VisualAge
    - ObjectStudio, Dolphin, Ambrai
    - GnuST, ST/X, S-Sharp
    - Gemstone/S, Envy



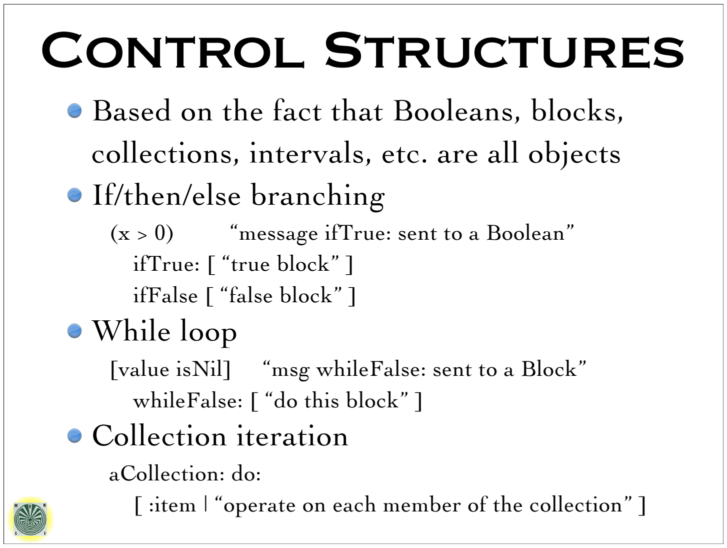
Smalltalk Elements

- The Language
  - Minimal, Uniform, Stable, Beautiful, Many Implementations, Compiles to VM language
- The Libraries
  - Standardized, Stable, Comprehensive, Consistent
- The IDE
  - Best-in-breed, Cross-platform
  - Extensible tools



The Smalltalk Language

- Uniform Message-passing metaphor
  - Everything (even int 3) is an object
- Unary messages
  - 3.2 sin, x negated, Date today printString
- Keyword Messages
  - x := anArray at: 1
  - anArray at: 1 put: anObject
  - aDisplayObject displayOn: aDevice at: aPoint clip: aRectangle mask: aMaskForm...
- The Protocol of Object
- Useful Short-hand
  - x @ y, key -> value



Control Structures

- Based on the fact that Booleans, blocks, collections, intervals, etc. are all objects
- If/then/else branching

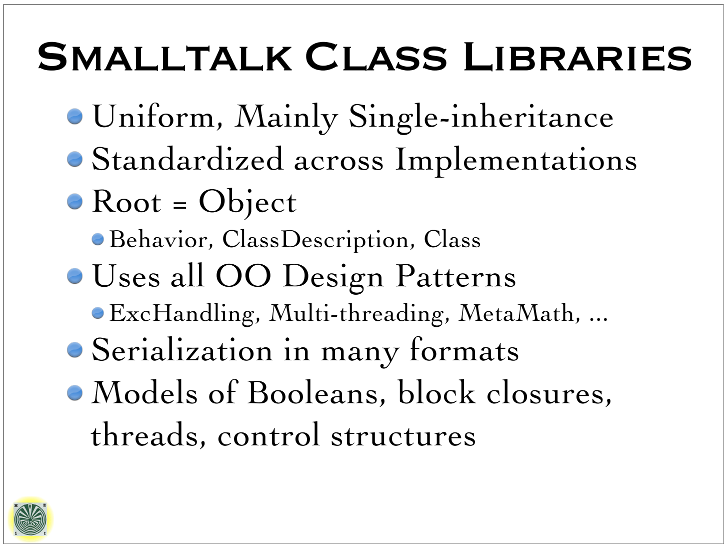
```
(x > 0)        "message ifTrue: sent to a Boolean"
   ifTrue: [ "true block" ]
   ifFalse [ "false block" ]
```

- While loop

```
[value isNil]    "msg whileFalse: sent to a Block"
   whileFalse: [ "do this block" ]
```

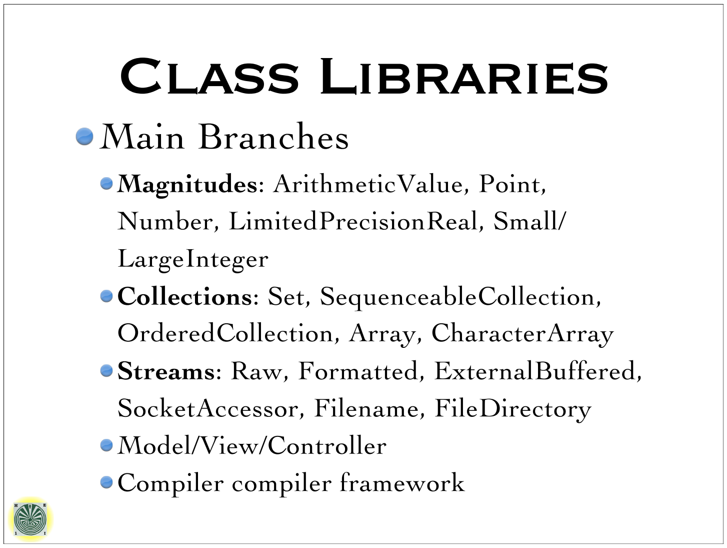
- Collection iteration

```
aCollection: do:
   [ :item | "operate on each member of the collection" ]
```



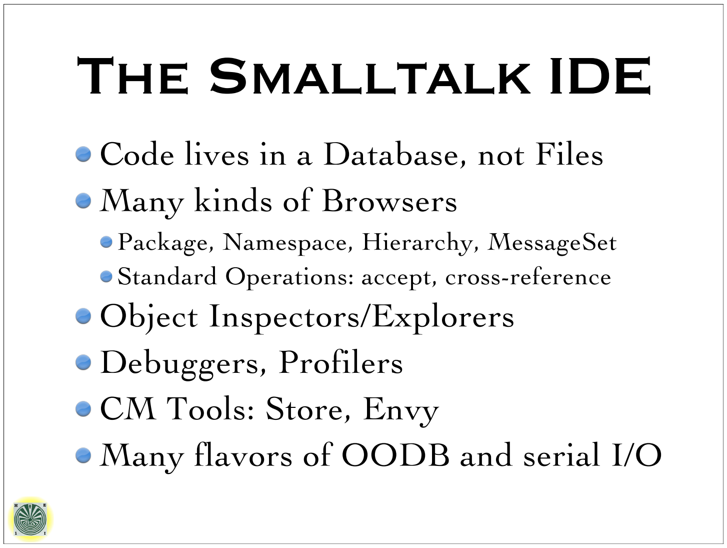
Smalltalk Class Libraries

- Uniform, Mainly Single-inheritance
- Standardized across Implementations
- Root = Object
  - Behavior, ClassDescription, Class
- Uses all OO Design Patterns
  - ExcHandling, Multi-threading, MetaMath, ...
- Serialization in many formats
- Models of Booleans, block closures, threads, control structures



Class Libraries

- Main Branches
  - **Magnitudes**: ArithmeticValue, Point, Number, LimitedPrecisionReal, Small/ LargeInteger
  - **Collections**: Set, SequenceableCollection, OrderedCollection, Array, CharacterArray
  - **Streams**: Raw, Formatted, ExternalBuffered, SocketAccessor, Filename, FileDirectory
  - Model/View/Controller
  - Compiler compiler framework



The Smalltalk IDE

- Code lives in a Database, not Files
- Many kinds of Browsers
  - Package, Namespace, Hierarchy, MessageSet
  - Standard Operations: accept, cross-reference
- Object Inspectors/Explorers
- Debuggers, Profilers
- CM Tools: Store, Envy
- Many flavors of OODB and serial I/O

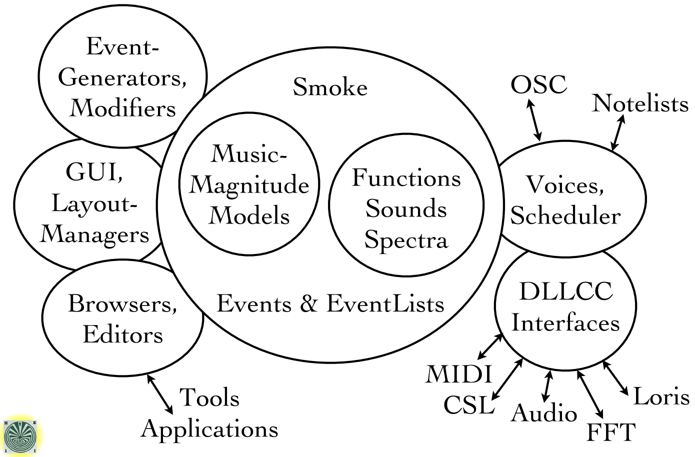
The Smalltalk Run-time

- Virtual machine based (like Java)
  - NCC, PIC, HotSpot, JITTER
- Garbage collection in VM
- Code and objects stored to virtual image file (~5-20 MB), like an APL workspace or Lisp core, objects held in threads
- Connects to DBMS, socket protocol adaptors, shell processes, other VMs, CORBA, XML/UDP...

Siren Set-up, Build

- Installing, Starting Smalltalk
  - Required Files
  - Dynamic Libraries
  - Useful Utilities (audio, MIDI, OSC, SC)
- Building Siren
- Using the Smalltalk Environment
  - Browsers, File Tools, Debuggers, Store
- Siren System Configuration
  - Loading User Data, Siren DBs
- Launcher, Utility, Transport Tools

Siren Overview



Smoke

- MusicMagnitudes, PitchClasses
- Events and EventLists
- EventGenerators, EventModifiers
- Voices, Scheduling, Performance
- Functions, Sounds, and Spectra

MusicMagnitudes

MusicMagnitudes

```
440 Hz asSymbol          --> 'a3' pitch
(1/4 beat) asMsec        --> 250 msec
#mf ampl asMIDI          --> 70 velocity
-16 dB asRatio value     --> 0.158489
('a4' pitch + 100 Hz) asMIDI        --> 73 key
'mp' ampl + 3 dB                    --> -4.6 dB
(1/2 beat) + 100 msec               --> 0.6 beat
```

Pitch expressions

```
N C augmentedFourth
N do flat intervalBetween: N sol
PitchClass D flat melodicMinorScale notes
(PitchClassChord new fromString: 'C aug9 dim5') notes
(HungarianMinor root: N fa) asPitchesInOctave: 2
```

Events

Creating Events Verbosely

```
MusicEvent duration: 1/4 pitch: 'c3' ampl: 'mf'
(MusicEvent dur: 1/4 pitch: 'c3') color: #green; accent: #sfz
```

Terse format: concatenation (with ',') of music magnitudes

```
440 Hz, (1/4 beat), 44 dB
(#c4 pitch, 0.21 sec, 64 velocity) voice: OSCVoice default
```

EventLists

Convenience Constructors

```
EventList named: 'melody'
 fromSelectors: #(pitch: duration: ampl:)  “selectors”
 values: (Array with: #(c d e f g)              “value array array”
     with: #(4 8 8 4 4) reciprocal  with: 1)

EventList random: 64                "make 64 notes"
 from: ((#dur: -> (50 to: 200)),    "duration range in msec"
     (#pitch: -> (36 to: 60)),      "pitch range in MIDI keys"
     (#ampl: -> (48 to: 120)),      "amplitude in MIDI vel"
     (#voice: -> (1 to: 1))         "play all on voice 1"
```

EventGenerators, -Modifiers

EventGenerators

```
Cluster  dur: 2.0 pitchSet: #(48 50 52 54 56) ampl: #ff
Roll length: 2 rhythm: 50 note: 60
Cloud dur: 6                  "lasts 6 sec."
     pitch: (48 to: 60)       "with pitches in this range"
     ampl: (80 to: 120)       "and amplitudes in this range"
     density: 5               "5 notes per sec"
DynamicSelectionCloud dur: 6
     pitch: #(#(48 50 52)     "starting/ending pitch sets"
             #(72 74 76) )
     ampl: #(60 80 120) density: 12
```

EventModifiers

```
roll := (Roll length: 3 rhythm: 150 note: 60) eventList.
decresc := Swell new function:
        (ExponentialFunction from: #((0 1 4) (1 0))).
decresc applyTo: roll
```

EGen Processing

```
“Create a list of tetrachords from a scale and play a shifting pattern on OSC”
chords := ((NeapolitanMinor root: N do) generateChordsPoly: 4 inOctave: 2).
chords := chords scrambled.                  “Scramble list of tetrachords”
list := OrderedCollection new.               “List for ((time -> chord) () ... )”
1 to: 7 do: [ :ind |                         "shift every 2nd up 2 octaves"
    ind even ifTrue: [list add: ((ind - 1) * 3 -> ((chords at: ind)
          “3 seconds per chord”     collect: [ :no | no + 24]))]
          ifFalse: [list add: ((ind - 1) * 3 -> (chords at: ind))]].
score := (ExtDynamicSelectionCloud           "now make a cloud from these"
    dur: 8  pitch: list  ampl: 60  density: 10) eventList.
score eventsDo: [ :ev |                      "plug in the properties for FM"
    ev inst: '/i1/pn'.       ev modIndex: 2.0.
    ev ratio: 1.02.          ev pos: 0.0]
```

Voices, Performance

```
(EventList named: ‘piece1/mvmnt1/sect1’)
      playOn: MIDIVoice default

strm := (Filename named: ‘output.sc’) writeStream.
vox := SuperColliderVoice newNamed: '1' onStream: strm.
vox play: anEventList.
strm close.

voice := OSCVoice default.
voice pMap: OSCVoice pMapForCSLSnd.     "plug in map"
voice send: '/i5/fi' args: (Array with:         "load a sound file"
                (SirenUtility findFile: 'moon.snd')).
anEventList voice: voice.
anEventList play
```

Functions, Sounds

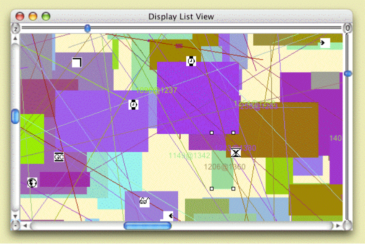
```
LinearFunction from: #((0 0) (0.5 1) (1 0))        "triangle envelope"
Function from: (((SHARCInstrument fromDir: 'tuba') samples at: #c3)
                        asWavetable: 1024)

SampledSound sweepDur: 10.0 rate: 44100 from: 10 to: 400 chans: 1
(SampledSound fromFile: 'unbelichtet.aiff') edit

(SampledSound sineDur: 1.0 rate: 44100 freq: 220 chans: 1)
      scaledByFunction: (ExponentialFunction default)

sin := (SampledSound sine: 1.0 rate: 44100 freq: 10) scaleBy: 0.8.
saw := (SampledSound saw: 1.0 rate: 44100 freq: 100) scaleBy: 0.1.
1 to: sin size do: [ :index |     "loop to do vector math on sound samples"
   sin at: index put: ((sin at: index) + (saw at: index))]
```

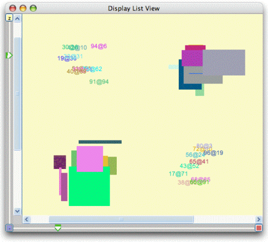
Graphics and GUI

- DisplayItems and DisplayLists
- DisplayListViews, and -Controllers
- Siren Views, Controllers, Inspectors
- Editor Models
- LayoutManagers
- EventList Views
- Sounds & Spectra: FFT & Loris

DisplayItems & -Lists

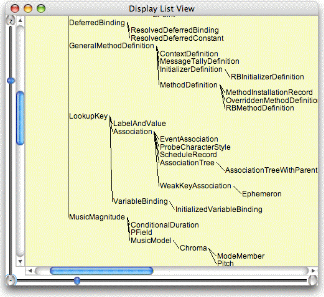
- DisplayItem classes
- DisplayList class
- DisplayListView/Controllers
- Pluggable widgets

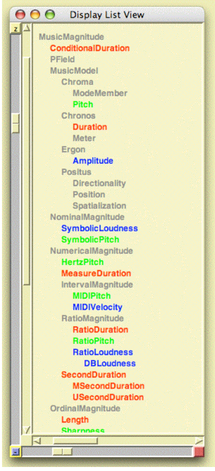




Layout-Managers

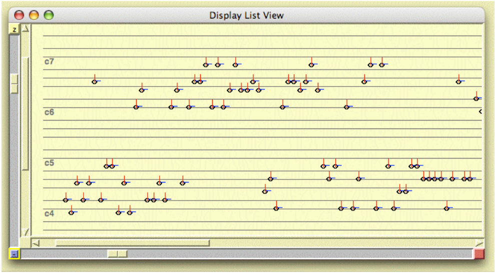
- Turn a data structure into a display list





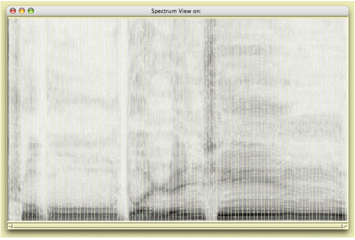
Siren Score Widgets

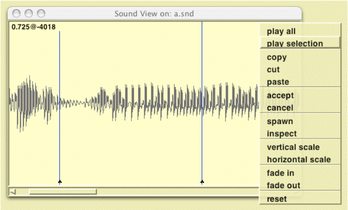
- Variations of PitchTimeLayoutManagers

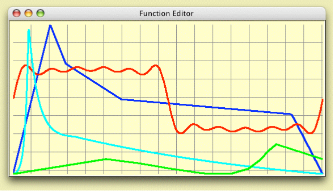


Siren Function Widgets

- Views and editors for functions, sounds, and spectra







I/O & Ext. Interfaces

- Setting-up, Testing External Interfaces
  - LibSndFile, PortAudio, PortMIDI, FFTW
- File, MIDI, OpenSoundControl I/O
- Sound, Soundfile I/O
- Interfaces to CSL, Loris and LPC
  - SWIG Interfaces and Coding
  - CSL, Loris Interfaces and Models

External Interfaces to C

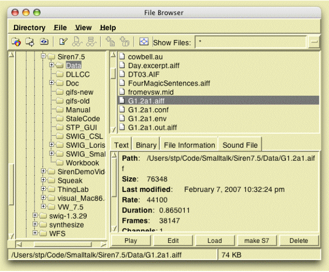
- Model objects use Interface objects, whose methods are references to C functions.

```
PortAudioInterface pa_play: out_buffer
            with: numChannels with: numFrames
    <C: int pa_play(unsigned short * out_buffer,
            unsigned int numChannels,
            unsigned int numFrames)>
    ^self externalAccessFailedWith: _errorCode
```

- Data shared between ST and C is copied to a special heap

Siren Tools

- Utility and Transport Views
- Extensions to FileList tool

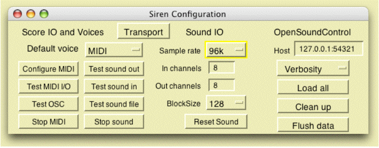

File Browser
Directory File View Help
Show Files: *
Text Binary File Information Sound File
Play Edit Load make S7 Delete

Siren7.5
Data
DLLCC
Doc
gifs-new
gifs-old
Manual
StaleCode
STP_GUI
SWIG_CSL
SWIG_Loris
SWIG_Smal
Workbook
SirenDemoVid
Squeak
ThingLab
visual_Mac86.
VW_7.5
swig-1.3.29
synthesize
WFS

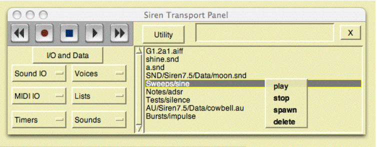
cowbell.au
Day.excerpt.aiff
DT03.AIF
FourMagicSentences.aiff
fromevsw.mid
G1.2a1.aiff
G1.2a1.conf
G1.2a1.env
G1.2a1.out.aiff

Path: /Users/stp/Code/Smalltalk/Siren7.5/Data/G1.2a1.aiff
Size: 76348
Last modified: February 7, 2007 10:32:24 pm
Rate: 44100
Duration: 0.865011
Frames: 38147
Channels: 1

/Users/stp/Code/Smalltalk/Siren7.5/Data/G1.2a1.aiff 74 KB



Default voice MIDI
Sample rate 96k
In channels 8
Out channels 8
BlockSize 128
Host 127.0.0.1:54321



G1.2a1.aiff
shine.snd
a.snd
SND/Siren7.5/Data/moon.snd
Sweeps/sine
Notes/adsr
Tests/silence
AU/Siren7.5/Data/cowbell.au
Bursts/impulse

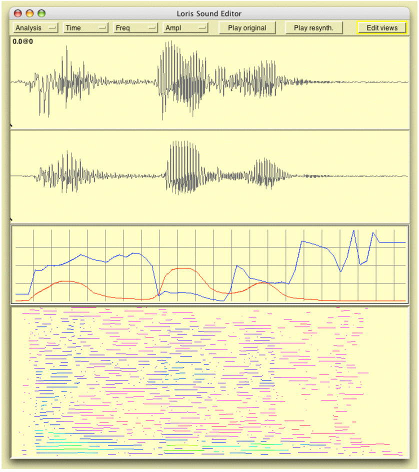
Applications, Editors

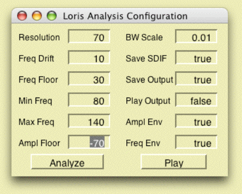
- Smalltalk Tool Extensions
- EventList/Generator Editors
- MIDI & OSC I/O
- Sound Processing, Mixing
- Feature Extraction and Score DBs
  - Scarlatti/Paleo
  - Bach/HAT
  - Wozzeck
  - TrTrees

Advanced Applications

- Loris Analysis and Morphing Tools
- CSL Interfaces
- LPC, Aubio File-based Interfaces
- CRAM Management

Loris Cross-Synthesis Editor





Resolution	70	BW Scale	0.01
Freq Drift	10	Save SDIF	true
Freq Floor	30	Save Output	true
Min Freq	80	Play Output	false
Max Freq	140	Ampl Env	true
Ampl Floor	-70	Freq Env	true

CSL Scripting

```
CSLGraph.sineWithADSR
     "Create and run a simple CSL DSP graph consisting of a
     Sine wave with an ADSR envelope."
     | env sin out |          "declare temp variables (optional)"
"Instrument: create a simple sine-with-envelope graph"
     env := ADSR dur: 2.0 att: 0.05 dec: 0.05 sus: 0.5 rel: 1.0.
     sin := Sine freq: 110 ampl: env.
"Create an IO object and plug in the sine"
     out := PAIO sRate: 44100 bSize: 1024 root: sin.
"Score: open the output, trigger the envelope, and start"
     out open.
     out start.
     env trigger.
"sleep a bit (3 sec)"
     3 seconds wait.
"shut down nicely"
     out clearRoot.
     out stop
```

Composition Examples

- *Day*
- *Leur Songe de la Paix*
- *Eternal Dream*
- *All Gates are Open*
- *Ora penso che il mondo...*
- *Jerusalem's Secrets*
- *Sensing/Speaking Space*

See also
Ritual and Memory
Sampler/Tour

Composition Examples

- For a 15-minute sampler/tour of my music and its score formats and composition tools, see

http://HeavenEverywhere.com/
RitualAndMemory/Tour

Evaluation, Summary

- Advantages
  - Integrated
  - Comprehensive
  - Cross-platform
- Disadvantages
  - Complex (ST-side, C-side)
  - Non-standard lang/env
  - Mixed-metaphor

Thanks, Acknowledgments

- Contributors
  - Mark Lentczner, Francois Pachet, Paul Alderman, Sekhar Ramakrishnan, etc.
- My Employers
  - Cadmus, Xerox PARC, STEIM, CCRMA, CNMAT, SICS, CREATE

For More Info.

- Siren Web Site: http://FASTLabInc.com/Siren
  - Down-loads
  - Docs
    - Papers (incl. ICMC 2007)
    - Workbook, Reference Manual
    - Pieces
  - Examples
  - Sample Data