

# HsAudio

Dan Brown   Jeffrey Yasskin

April 25, 2005

## Audio Processing Language

Ideal Language

Implementation

## Building an Audio Player

From the Processing Language?

Player Language

Demo

# Audio Processing Language

# Ideal Audio Combinator Language

## Playable

- ▶ *duration* : *Playable* → *Time*
- ▶ *play* : *Playable* → *Action*

# Ideal Audio Combinator Language

## Playable

- ▶ *duration* : *Playable* → *Time*
- ▶ *play* : *Playable* → *Action*
- ▶ *AudioFile* : *Path* → *Playable*
- ▶ *AudioStream* : *URL* → *Playable*

# Ideal Audio Combinator Language

## Playable

- ▶ *duration* : *Playable* → *Time*
- ▶ *play* : *Playable* → *Action*
- ▶ *AudioFile* : *Path* → *Playable*
- ▶ *AudioStream* : *URL* → *Playable*
- ▶ *sequence* : *Playable* → *Playable* → *Playable*
- ▶ *mix* : *Playable* → *Playable* → *Playable*

# Ideal Audio Combinator Language

## Playable

- ▶ *duration* : *Playable* → *Time*
- ▶ *play* : *Playable* → *Action*
- ▶ *AudioFile* : *Path* → *Playable*
- ▶ *AudioStream* : *URL* → *Playable*
- ▶ *sequence* : *Playable* → *Playable* → *Playable*
- ▶ *mix* : *Playable* → *Playable* → *Playable*
- ▶ *slice* : *Time* → *Playable* → (*Playable*, *Playable*)
- ▶ *loop* : *Playable* → *Playable*

# Ideal Audio Combinator Language

## Playable

- ▶ *duration* : *Playable* → *Time*
- ▶ *play* : *Playable* → *Action*
- ▶ *AudioFile* : *Path* → *Playable*
- ▶ *AudioStream* : *URL* → *Playable*
- ▶ *sequence* : *Playable* → *Playable* → *Playable*
- ▶ *mix* : *Playable* → *Playable* → *Playable*
- ▶ *slice* : *Time* → *Playable* → (*Playable*, *Playable*)
- ▶ *loop* : *Playable* → *Playable*
- ▶ ...

# Axioms

*empty* : *Playable*

*silence* : *Playable*

*duration empty* = 0

*duration silence* =  $\infty$

*sequence* is associative

*empty* 'sequence' *a* = *a*

*a* 'sequence' *empty* = *a*

*silence* 'sequence' *a* = *silence*

## Axioms (cont.)

*mix* is associative, commutative

$$a \text{ 'mix' } a = a$$

$$\text{empty 'shortMix' } a = \text{empty}$$

$$\text{empty 'longMix' } a = a$$

$$\text{silence 'shortMix' } a = a$$

$$\text{silence 'longMix' } a = a \text{ 'sequence' silence}$$

## Axioms (cont.)

*sequence (slice t a) = id*

*duration (loop a) =  $\infty$*

*loop = fold sequence empty  $\circ$  repeat*

# Implementation: Reality Sets In

- ▶ GStreamer is a pipeline-based C library
- ▶ Ideal language based on combinators

# Implementation: Reality Sets In

- ▶ GStreamer is a pipeline-based C library
- ▶ Ideal language based on combinators
- ▶ Fortunately, pipeline prefixes are composable

# The Real Language (so far...)

## Now

- ▶ *AudioFile*
- ▶ *Mix*
- ▶ *toPipeline* and *play*

# The Real Language (so far...)

## Now

- ▶ *AudioFile*
- ▶ *Mix*
- ▶ *toPipeline* and *play*

## Soon

- ▶ *Sequence*

# The Real Language (so far...)

## Now

- ▶ *AudioFile*
- ▶ *Mix*
- ▶ *toPipeline* and *play*

## Soon

- ▶ *Sequence*

## Ever?

- ▶ *slice*

# Building an Audio Player

# Another Language

Different from audio processing language

- ▶ Events
- ▶ First-class actions
- ▶ Interaction

Re-uses little of audio processing language

# Another Language

Different from audio processing language

- ▶ Events
- ▶ First-class actions
- ▶ Interaction

Re-uses little of audio processing language

Re-uses much of its implementation

# Playlist not a Sequence

## Playlists

- ▶ Contain player actions
- ▶ Are mutable
- ▶ Are navigable

Sequences are primitive, only sequence contents

Need both simple sequences and versatile playlists

- ▶ Sequences for processing
- ▶ Playlists for players

# Playlist not a Sequence

## Playlists

- ▶ Contain player actions
- ▶ Are mutable
- ▶ Are navigable

Sequences are primitive, only sequence contents

Need both simple sequences and versatile playlists

- ▶ Sequences for processing
- ▶ Playlists for players

Players + processing language = ?

# Audio Player Language

## Actions

- ▶ *play* : Action
- ▶ *pause* : Action
- ▶ *stop* : Action
- ▶ *quit* : Action
- ▶ *add* : Playable  $\rightarrow$  Action
- ▶ *next* : Action
- ▶ *previous* : Action
- ▶ *jump* : Int  $\rightarrow$  Action
- ▶ *seek* : Time  $\rightarrow$  Action
- ▶ *playlist* : Query Playlist

# Implementation

# Demo