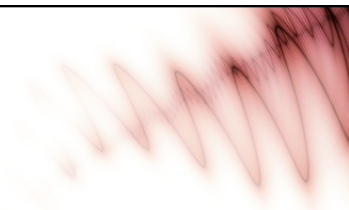


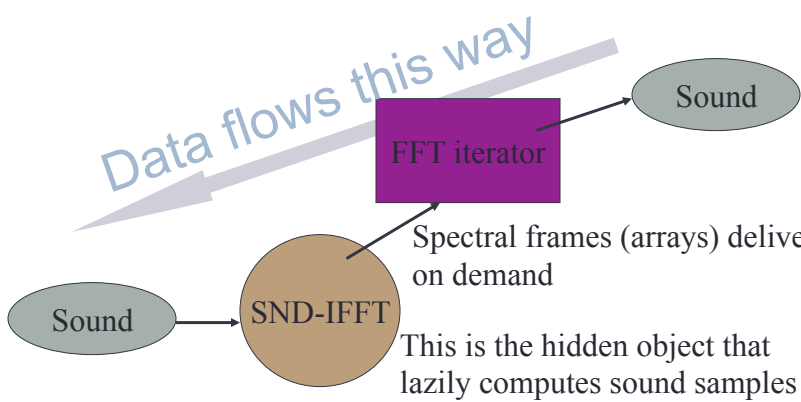
SPECTRAL PROCESSING

Using SAL to operate on spectra

ICM Week 8 Copyright © 2002-2013 by Roger B. Dannenberg 1



From Sound to Spectra



Data flows this way

Sound

FFT iterator

Spectral frames (arrays) delivered on demand

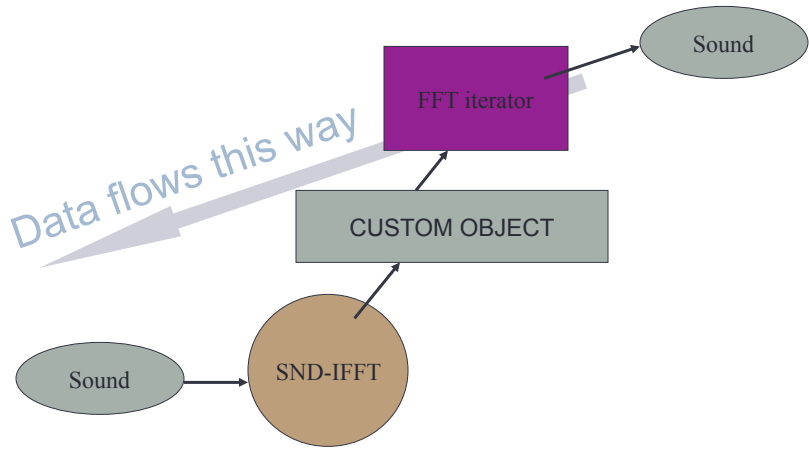
Sound

SND-IFFT

This is the hidden object that lazily computes sound samples

ICM Week 8 Copyright © 2002-2013 by Roger B. Dannenberg 2

OOP vs SAL

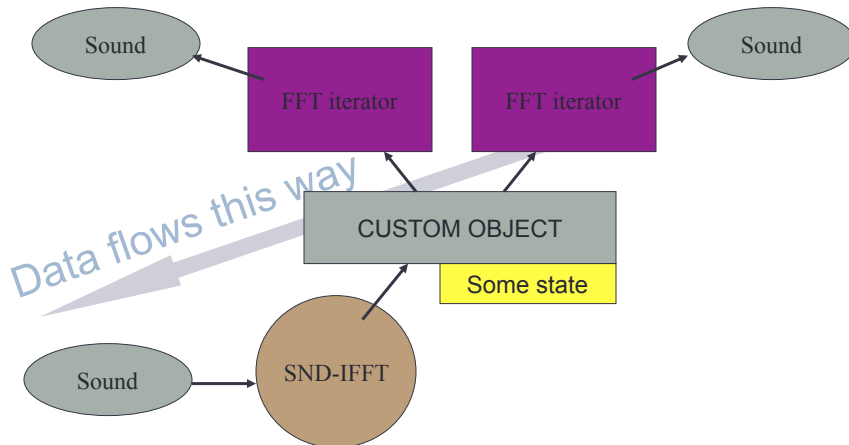


ICM Week 8

Copyright © 2002-2013 by Roger B. Dannenberg

3

OOP vs SAL

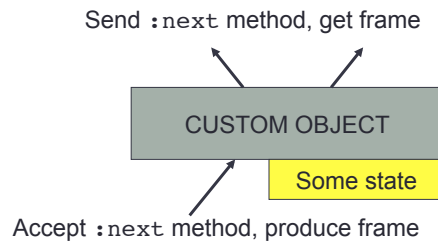


ICM Week 8

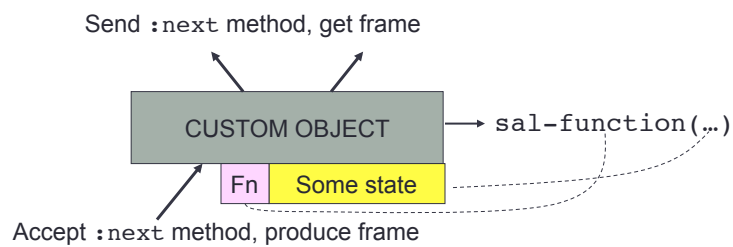
Copyright © 2002-2013 by Roger B. Dannenberg

4

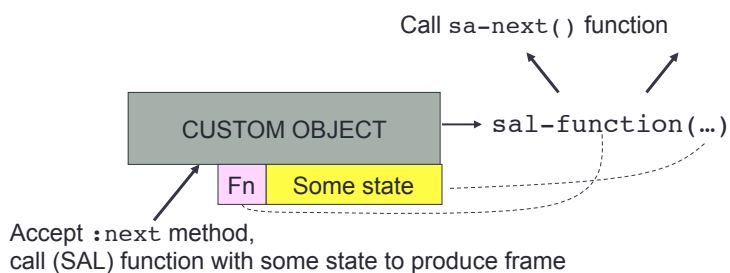
The Object Behavior We Need



How To Get Object Behavior From SAL



How To Get Object Behavior From SAL (2)



Template for Spectral Processing (1)

```
set sa = sa-init(input: "./rpd-cello.wav",
                fft-dur: 4096 / 44100.0,
                skip-period: 512 / 44100.0,
                window: :hann)
```

Template for Spectral Processing (2)

```

set sp = sp-init(sa, quote(processing-fn), 0, 0)

```

Spectral processing object Spectral analysis object SAL Function Initial State

```

function processing-fn(sa, frame, p1, p2)
begin
  ... Process frame here ...
  set frame[0] = 0.0 ; simple example: remove DC
  return list(frame, f(p1), g(p2)) ; state change
end

play sp-to-sound(sp)

```

Simple analysis/synthesis examples

- See `spectral-process.sal`
 - Note: requires `spectral-process.lsp` and `spectral-analysis.lsp` as well.

Cross-Synthesis, Morphs, etc.



- In general, combine features from two sounds
- Common approach: separate sounds into *excitation* and *filter* parts.
- Combine filter of one signal with excitation of the other, e.g.
 - Vocal tract filter applied to noise, orchestra, etc.
 - Cello body applied to woodwind sound
- See `spectral-process.sal`