NYQUIST, SAL, LISP

Getting started with Nyquist
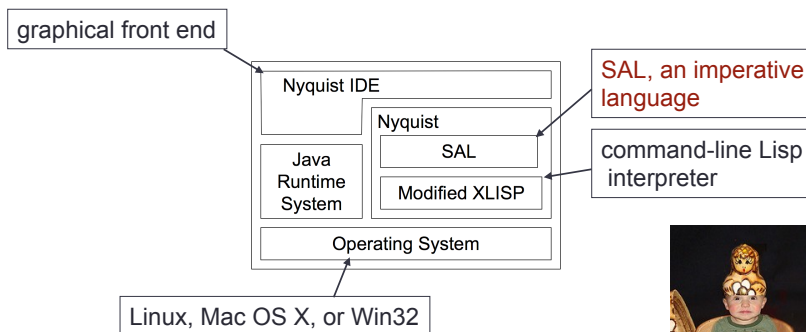
Nyquist, SAL, Lisp

graphical front end

Nyquist IDE

Nyquist

SAL

Java Runtime System

Modified XLISP

Operating System

SAL, an imperative language

command-line Lisp interpreter

Linux, Mac OS X, or Win32

From Cynose's photostream, Flickr.com, Feb 24, 2007

Nyquist: Top-Down

- NyquistIDE written in Java (requires Java runtime)
- interacts through sockets with SAL, written in XLISP
- XLISP is interpreted, written in C
- C is of course compiled to your native instruction set
- But there's more:
  - XLISP is extended with signal processing primitives
  - Written as high-level specifications (see Nyquist Ref. Manual)
  - Translated by XLISP program (tran.lsp) into C
- And more ..
  - score data structures are interpreted by a built-in function (`timed-seq`) that calls on the XLISP `eval` function.

Read-Eval-(Print) Loop

- You enter commands into SAL
- SAL reads the command and compiles it to XLISP
- XLISP evaluates the compiled command
- This may or may not generate output

Some Examples

- `play pluck(c4)`

- `play pluck(c4) ~ 3`

- `load "pianosyn"`

- `play piano-note(5, fs1, 100)`

- `play osc(c4)`

- `play osc(c4) * osc(d4)`

- `play noise() * env(0.05, 0.1, 0.5, 1, 0.5, 0.4)`

Some SAL Commands

- **print** *expression* - evaluate and expression and print the result
- **exec** *expression* - evaluate expression but *do not* print the result
- **play** *expression* - evaluate and expression and play the result, which must be a SOUND
- **set** *var = expression* - set a variable