Various IRCAM free software: jMax and OpenMusic

Francois Dechelle

dechelle@ircam.fr,

IRCAM, Paris, France

Free software at IRCAM

- jMax, OpenMusic, SDIF
- jMax GPL’ed in 1999 (first IRCAM free software)
- “Free software” team since mid-2002
- Partner of the AGNULA project

jMax

- yet another implementation of MAX: PD, MAX/MSP
- but uses a different architectural approach
- full separation between GUI (JAVA) and FTS audio engine (C)
- FTS is a *library*

jMax + JACK

- FTS scheduler runs inside the JACK callback
- callback is not yet RT safe: does a select() :-(
- configuration panel allows to connect to already running applications

jMax ladspa object

- use a LADSPA plugin as a patch object
- plugin ports (audio and control) are inlets/outlets
- plugin is given ala applyplugin
- no GUI for plugin

jMax LADSPA plugin

- run a patch as a LADSPA plugin
- audio input/output are audio ports
- controls ports are not implemented yet
- loads the patch in plugin instantiate()
- runs the FTS scheduler in plugin run()

jMax Python/GTK GUI

- pre-alpha state
- motivations:
  - Java is not free, so it is not packaged in AGNULA, so replace it
  - Python is better for prototyping GUIs

jMax client/server IPC

- simple protocol: int, float, string, object
- efficient w.r.t. memory allocation (Java/GC, real time...)
- uses either a socket or a pipe as transport layer

jMax client library

- provides a simple API on top of the protocol
- implemented in Java, C++ and Python
- Python code:

```
fts = Fts( "/usr/bin/fts")
cnt = FtsSocketConnection()
obj = FtsObject(cnt, None, "osc~")
obj.send( "frequency", 440.0)
```

OpenMusic

- a visual programming language based on CommonLisp/CLOS
- icon oriented, uses extensively drag&drop
- built-in visual control structures
- classes and libraries for music composition

OpenMusic concepts

- *patches*: programmation units
- *classes*: prototypes for objects
  - once in a patch, a class becomes a *factory*
  - instances often associated with an *editor*
- *functions* and *methods*: LISP
- *maquettes*: a special kind of patch with a time dimension

OM Linux

- OM development base platform: Macintosh + Digitool MCL
- Linux port by Gerardo Sarria and Jose Diago (Universidad Javeriana, Cali, Columbia)
- based on OpenMusic version 3.5
- uses CMUCL

OpenMusic graphical layer

- implementation of MCL graphical toolkit on top of GTK
- uses CMUCL GTK bindings
- not complete yet: editors are missing
- license issue ?

OpenMusic TODO

- merge with Macintosh code (version 4.7)
- audio layer: play sound files
- precise scheduling for maquettes
- MIDI layer: record and play MIDI sequences
- on Macintosh, uses MidiShare (GRAME)

URLs

- jMax:

  `http://www.ircam.fr/jmax`

- jMax sourceforge:

  `http://sourceforge.net/projects/jmax`

- OpenMusic:

  `http://www.ircam.fr/openmusic`

- OpenMusic sourceforge:

  `http://sourceforge.net/projects/ircam-openmusic`