

# CLAM, una llibreria lliure per àudio i música

Xavier Amatriain ([xamat@iua.upf.es](mailto:xamat@iua.upf.es))

Pau Arumí ([parumi@iua.upf.es](mailto:parumi@iua.upf.es))

David García ([dgarcia@iua.upf.es](mailto:dgarcia@iua.upf.es))

Grup de Tecnologia Musical

Institut Universitari de l'Audiovisual

Universitat Pompeu Fabra

26 de maig de 2003

## 1 Què és CLAM?

CLAM (C++ Library for Audio and Music) és un framework que ofereix als desenvolupadors solucions de disseny i un conjunt de components reusables per construir aplicacions musicals i d'àudio i per construir sistemes de recerca en l'àmbit del processat del senyal.

## 2 A qui pot interessar CLAM?

CLAM és adient per qualsevol que vulgui desenvolupar una aplicació, multiplataforma, eficient i ben estructurada, d'àudio o de música. En comparació amb altres llibreries de processat, CLAM destaca especialment quan el processat de l'aplicació es realitza en els dominis espectral o semàntics.

## 3 Per què vam desenvolupar CLAM?

CLAM ha estat desenvolupat pel Grup de Tecnologia Musical (MTG) de l'Institut Universitari Audiovisual (IUA) de la Universitat Pompeu Fabra (UPF). La recerca de l'MTG és reconeguda internacionalment en àrees com ara la síntesi i processat de la veu cantada, i extracció de característiques d'alt nivell de l'àudio.

Fa uns anys (vora el 2000), el MTG havia desenvolupat ja algunes aplicacions molt interessants relacionades amb les tècniques d'anàlisi i síntesi espectral. L'evolució d'aquestes

aplicacions era específica per cadascuna d'elles i sovint més guiada per l'experimentació que per criteris de reusabilitat i mantenibilitat.

En resum, cada aplicació havia de començar pràcticament de zero i era difícil mantenir o reusar el codi existent, o propagar les innovacions que es fessin a un projecte cap als altres projectes.

A més, tot i que, algunes aplicacions eren distribuïdes públicament de forma gratuïta en binaris, sense codi font públic, era molt difícil obtenir contribucions o feedback d'altres grups d'investigació externs a l'MTG.

CLAM va sorgir com a forma de facilitar aquest intercanvi i transferència de entre els diferents projectes i per proveir de conceptes de disseny, extrets i depurats a partir de l'experiència adquirida, que facilitessin la construcció d'aplicacions.

CLAM encara està evolucionant, però ja s'ha fet servir exitosament en múltiples projectes del MTG i alguns altres de l'exterior. En aquests projectes, CLAM ha suposat una millora substancial en el temps de desenvolupament, en la claretat i mantenibilitat del codi, i, a més, ha aportat, a aquestes aplicacions, algunes funcionalitats amb poc esforç que d'altra manera no haurien estat implementades.

## 4 Com es distribueix CLAM?

CLAM es va començar desenvolupant com a projecte intern per cobrir les necessitats pròpies del MTG, però, de seguida va ser clar que podria ésser interessant fora del nostre propi entorn.

A l'Abril de 2002, va començar el projecte europeu IST Agnula. L'objectiu d'aquest projecte, finançat per la Unió Europea, és compilar un parell de distribucions de GNU/Linux especialitzades en aplicacions d'àudio. Una de les distribucions (RedHMudi) està basada en RedHat i l'altra (DeMuDi) basada en Debian. CLAM ha estat incorporat a aquest projecte, i, en fer-ho, ha esdevingut distribuïble lliurement sota llicència GNU-GPL.

CLAM, és doncs, software lliure, i qualsevol aplicació pot fer-la servir sempre que també estigui llicenciada sota la GNU-GPL. CLAM també fa servir un esquema de doble llicència que permet, previ acord amb el MTG, desenvolupar aplicacions amb CLAM sota altres llicències.

## 5 Què inclou CLAM?

CLAM ofereix, per un costat, components a punt per fer-se servir per construir aplicacions noves només per composició. Per un altre costat, ofereix una arquitectura oberta que permet construir per extensió nous conceptes que es podrien incorporar més tard a la llibreria.

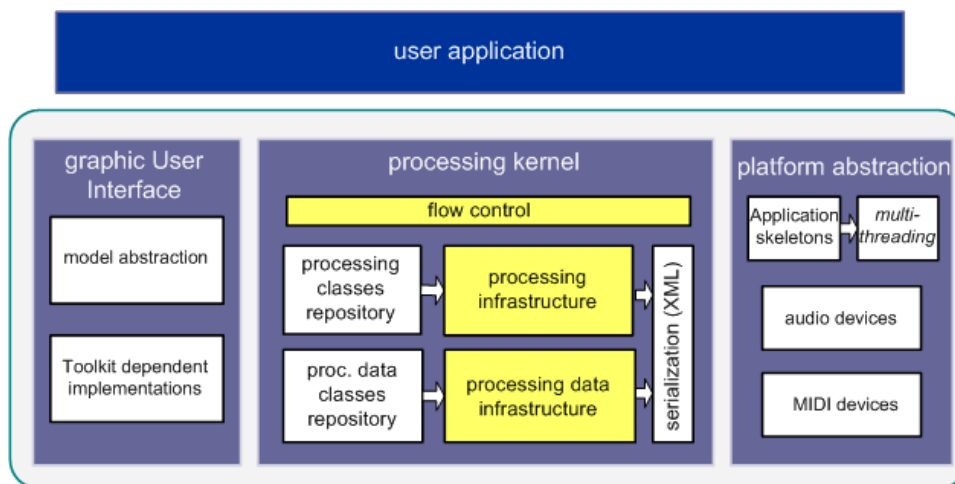


Figura 1: Arquitectura de CLAM

La major part dels components de CLAM són classes de processat (Processing's) i les estructures de dades que prenen com a entrada i sortida (ProcessingData's). Alguns exemples de classes de processat que ja inclou CLAM són les FFT/IFFT, SMSAnalysis, SMSSynthesis, diverses transformacions espectrals, processat MIDI, generadors d'ona, filtres en domini temporal...

Fora del processat, CLAM també incorpora una caixa d'eines prou nutrida i independent de plataforma que permet cobrir àrees tan comuns i laborioses pel desenvolupament d'aplicacions d'àudio com ara la lectura i escriptura d'àudio i MIDI a dispositius digitals o a fitxer, la programació multifils, la visualització interactiva de dades, serialització XML i SDIF...

També ofereix facilitats per construir plug-ins Ladspa a GNU/Linux o VST a M\$-Windows.

## 6 Aplicacions desenvolupades usant CLAM

Hi ha diversos projectes de l'MTG i de fora que fan servir CLAM. Alguns exemples són:

**SMS Analysis/Synthesis (lliure):** És una reimplementació amb CLAM d'una eina de l'MTG, SMSTools, que donava accés experimental a les tècniques de processat per models espectrals (SMS). Molts dels projectes del MTG es basen en aquesta tecnologia.

**SALTO (lliure):** És un sintetitzador d'alta qualitat d'instruments de vent implementat per software amb tècniques SMS. Actualment estan implementats el saxo i la trompeta.

**Spectral Delay (lliure):** És un banc de filtres multibanda en el que podem retardar cada banda per separat. És un exemple típic de com les facilitats de CLAM pel processat

espectral poden simplificar molt el desenvolupament.

**Time Machine (propietari):** És un algorisme de *time stretching* d'alta qualitat que està fent-se servir actualment en productes comercials.

**CUIDADO (mixt):** És un projecte europeu que es centra en l'extracció de descriptors d'alt nivell de l'àudio (rítmics, melòdics...) i la seva utilització per fer manipulacions sobre l'àudio original. CLAM inclou ja gran part del codi del projecte que ha esdevingut lliure.

**Vocal Processor (propietari):** És un plug-in VST que permet processar la veu cantada tot fent canvis de pitch, de timbre, afegint harmonitzacions... tot a temps real.

## 7 Model de desenvolupament

Tot i que el desenvolupament de CLAM al seus inicis ha estat poc obert a l'exterior, les eines i els procediments han estat des de bon començament els propis del software lliure.

Entre les eines, cal destacar les que han facilitat la comunicació entre desenvolupadors com ara:

- el CVS i els seus front-end gràfics entre els quals destaca LinCVS per la seva facilitat d'ús sobretot en la gestió de branques,
- el sistema de bugtracking (Mantis),
- les llistes de correu (Mailman), i,
- Doxygen una eina semblant a JavaDoc que serveix per generar documentació del codi i és molt útil per inspeccionar relacions creuades i dependències.

Un cop incorporat CLAM a Agnula, el desenvolupament s'ha obert a l'exterior, i hem començat a rebre cada cop més aportacions dels usuaris, tant en codi com en suggerències.

Les suggerències i els patches dels usuaris de CLAM han estat molt valuoses. Per una banda, ens ha permès veure quines funcionalitats eren més necessàries, quines calia modificar i quines altres treure. Això lligat amb el fet de les entregues petites i sovintejades ha donat un valor al framework que altrament no tindria. Per l'altre banda al ser els usuaris de CLAM també desenvolupadors, han aportat informes d'error prou detallats i a vegades fins i tot les solucions.

Darrerament, hem començat a aplicar al desenvolupament tècniques del que es coneix per eXtreme Programming. Els resultats de la seva aplicació han estat molt bons, tant en la velocitat de desenvolupament com en la qualitat del codi generat. De les tècniques emprades ens agradaria remarcar tres:

**Test Driven Development:** es tracta d'un nou enfocament en com s'escriu el codi: es comença escrivint el test de la funcionalitat que cal incloure i després s'implementa la funcionalitat. A més tots els testos es passen constantment. Programant amb aquest estil s'assegura:

1. que el codi estarà cobert per testos (unitaris) de regressió. És a dir, immediatament es coneix quan alguna modificació del codi ha canviat inesperadament algun comportament del programari.
2. els testos d'abast tant petit, quan fallen, donen una informació molt concreta de què i perquè està fallant,
3. dissenyar per que es pugui testejar, fa que el disseny tingui bones qualitats: components altament cohesionats i desacoblament entre ells.

**Pair Programming:** consisteix en portar el límit la idea de la revisió humana del codi (peer-reviewing). Com diu el nom de la tècnica es programa amb una parella asseguda davant el mateix ordinador. Així la revisió és constant i alhora permet que un es focalitzi més en el detall (el que té el teclat a les mans) mentre l'altre pot pensar en el proper pas a fer.

**Refactoring:** és el procés de modificar el codi sense modificar el seu comportament extern, amb l'objectiu de fer més fàcil la incorporació de noves característiques al codi. Molts refactorings es basen en la reduir la duplicació de codi.

## 8 Conclusions

CLAM és una llibreria lliure de qualitat que facilita l'aparició, als entorns lliures, de eines multimèdia basades en tecnologies avançades. Algunes d'aquestes eines ja s'han desenvolupat o estan sent desenvolupades per part de l'MTG i esperem que, sobretot amb la incorporació de CLAM a diverses distribucions de GNU/Linux, en el marc del projecte europeu Agnula IST, s'incrementi substancialment el seu ús fora del nostre grup.

## 9 Agraïments

El treball aquí reflexat ha estat finançat en part pels projectes Europeus IST AGNULA i CUIDADO.