

Extracció d'acords amb l'Anotador de Música de CLAM

David García
Music Technology Group
Univ. Pompeu Fabra
dgarcia@iua.upf.edu

Pau Arumí
Music Technology Group
Univ. Pompeu Fabra
parumi@iua.upf.edu

Xavier Amatriain
CREATE
Univ. of California Santa Barbara
Santa Barbara, CA, USA
xavier@create.ucsb.edu

Resum

Aquest article presenta una eina que hem desenvolupat al Grup de Tecnologia Musical¹ de la Universitat Pompeu Fabra². Es tracta d'una eina extensible d'extracció i edició d'informació de la música anomenada *CLAM Music Annotator* [2]. La darrera versió incorpora un extractor d'acords que amplia l'interès de l'eina més enllà de l'ús científic pel que estava pensat. Aquest extractor d'acords pot ser molt útil per als músics i aficionats que fins ara havien de treure els acords de les cançons d'oïda amb prova i error. L'article explica els principis de funcionament del programa i com es pot adaptar a diferents escenaris. També s'explica el cas concret de l'extracció d'acords, es fa una discussió tècnica del disseny, tecnologies emprades i lliçons apreses i, finalment, es comenta quin sera el camí probable que pendrà l'eina en el futur.

1 Context

El *CLAM Music Annotator* es distribueix com una de les aplicacions d'exemple del framework CLAM[1]. CLAM és un framework per construir aplicacions d'àudio i música desenvolupat també pel Grup de Tecnologia Musical i que rep, actualment, financiació de la Generalitat de Catalunya.

L'aplicació s'ha desenvolupat en el marc del projecte europeu SIMAC³. Ha estat un projecte amb dos anys i mig de durada que ha finalitzat recentment. El seu objectiu era construir uns

¹<http://mtg.upf.edu>

²<http://www.upf.edu>

³<http://www.semanticaudio.org>

prototips que demostressin la utilitat pràctica dels darrers avenços en el camp de la Recuperació d'Informació de la Música[7]. El CLAM Music Annotator és una part d'un d'aquests prototips que es pot descarregar lliurement des de la pàgina de CLAM⁴ El projecte també ha donat altres resultats interessants com ara l'aplicació web⁵ *Foafing The Music*[3].

2 Què és

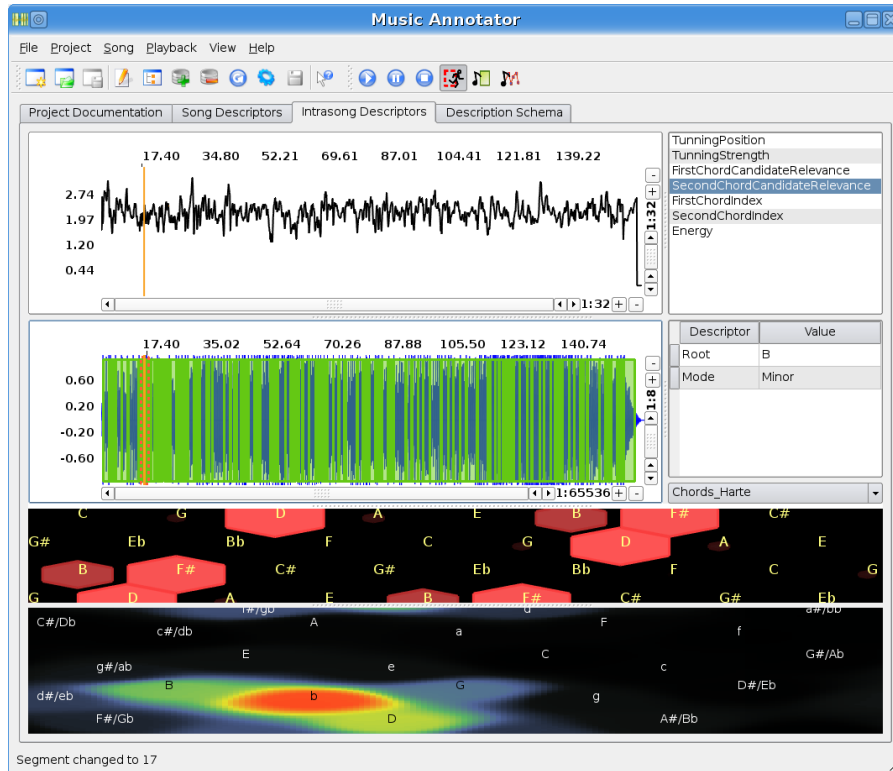


Figura 1: Extracció d'acords amb el Music Annotator

El CLAM Music Annotator⁶ és una eina de recerca que extreu informació de la música, la visualitza convenientment, i permet editar-la. Les seves funcionalitats estan adreçades a la recerca en el camp de la Recuperació d'Informació de la Música (*Music Information Retrieval*). Aquest camp de recerca es dedica a l'extracció automàtica d'informació del contingut de l'àudio per poder tractar-ne grans col·leccions de forma automatitzada. L'utilitat d'un programa d' anotació en aquest camp es poder generar dades a mà que serveixin d'entrenament o de validació per als algorismes d'extracció i també supervisar i corregir, si cal, la seva sortida.

Aquestes dades normalment poden ser:

- atributs a nivell de cançó, com ara el tempo, la clau, la tonalitat...

⁴<http://clam.iua.upf.edu/download.html>

⁵<http://foafing-the-music.iua.upf.edu>

⁶http://iua-share.upf.edu/wikis/clam/index.php/Music_Annotator

- atributs instantànis, que caracteritzen instants molt petits, com ara, l'energia, la transformada de fourier, el to predominant...
- atributs associats a segments temporals, com ara el to o la velocitat d'una nota, o l'arrel i el mode d'un acord, l'estructura de la peça.

Fins ara, era molt comú fer servir, per aquestes tasques d'anotació, el programa WaveSurfer[10]. Aquest programa⁷, però, tot i ser lliure i extensible, és molt inestable i té diverses limitacions en l'edició en quan a flexibilitat i ergonomia. Alguns científics fins i tot, donades aquestes limitacions, preferien fer servir programari genèric d'edició d'àudio, no pensat per anotar, com ara Audacity, SoundForge, LogicAudio... [8] El nostre grup va contribuir al WaveSurfer amb diverses millores [5], però, pel fet de estar programat en Tcl/Tk, fa molt difícil fer-ne contribucions o mantenir-ho.

És per això que vam decidir d'oferir una eina més flexible i potent fent servir tecnologies més comuns i àgils com són el llenguatge C++ i el *toolkit* Qt. El Music Annotator de CLAM ofereix prou flexibilitat com per adaptar-se a l'estructura de la informació que estem extraient, que es defineix amb un esquema de dades en XML. També es poden incorporar nous tipus de dades, vistes, interfícies d'edició i auralitzacions

Tot i que aconseguim amb això fer més còmode el procés d'anotació, no deixa de ser una tasca molt feixuga; anotar a mà només unes poques cançons requereix molt de temps. A més, per que l'entrenament o la validació dels algorismes siguin fiables, cal tenir un nombre considerable d'anotacions manuals i, és convenient que la qualitat d'aquestes hagi estat verificada com a mínim per una segona persona. Les anotacions, com el programari, són un bé intangible; es poden compartir sense haver de repartir. El *CLAM Music Annotator* es pot fer servir en combinació amb un programari web, el *Barebone Of Collaborative Annotation* (BOCA), que serveix per coordinar campanyes d'anotació col.laboratives entre diversos centres de recerca.

3 Com funciona

3.1 Definint el projecte

Per fer servir l'Annotator, cal definir un projecte amb un schema de dades, una comanda extractora i una llista de cançons.

L'esquema de dades és un arxiu XML que descriu els atributs amb els que es treballa. Indica quin és l'àmbit (canço, nota, estrofa, instant...) de cada atribut, i quin és el seu tipus. Els tipus suportats per l'Annotator són ampliables. D'entrada s'ofereixen alguns tipus que permeten una potència mínima: els tipus bàsics (enters, reals, enumeracions, text i arrays o llistes dels anteriors) i tipus especials d'atribut que permeten enllaçar i definir àmbits per a altres atributs. Per exemple, definint un atribut que sigui una divisió en segments temporals dels compasos o de les notes, s'està definint l'àmbit dels atributs associat amb els compasos

⁷<http://www.speech.kth.se/wavesurfer/>

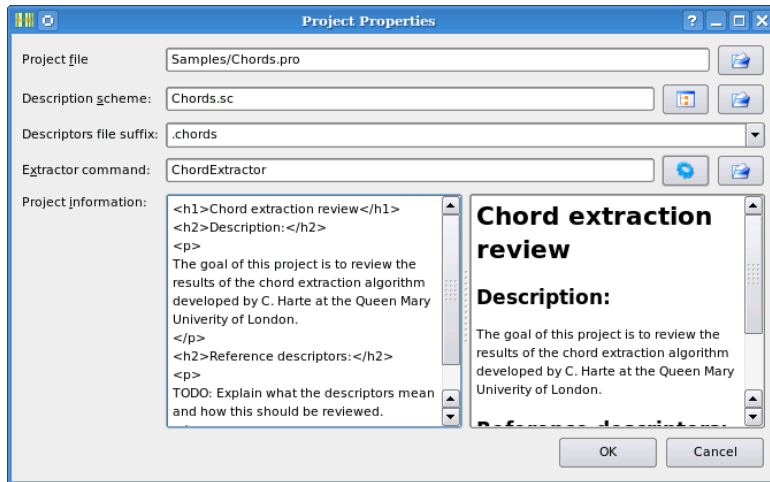


Figura 2: Propietats d'un projecte

o les notes. O, per exemple, les divisions equidistants que estableixen l'àmbit pels atributs instantanis que es donen cada pocs milisegons.

Els tipus de l'Annotator són tipus enriquits. Per exemple, els valors numèrics poden tenir o no limits, unitats... i quan definim un array podem establir què significa cada element associant-ho a etiquetes textuais o a una escala numèrica.

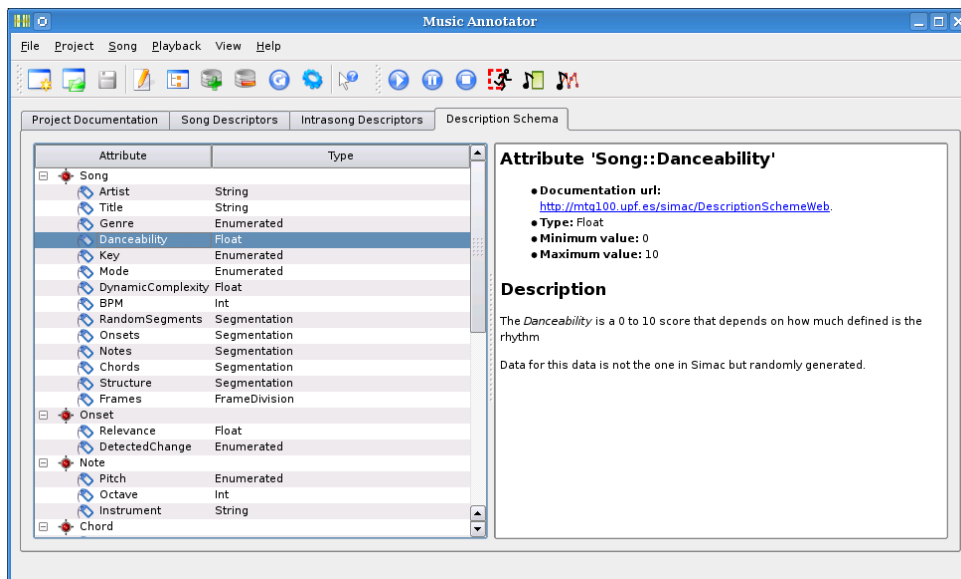


Figura 3: Explorador de l'esquema de descripció

El projecte també pot ser el resultat de baixar-se una tasca del BOCA. En aquest cas és especialment útil la documentació que es pot adjuntar al projecte donat que permet donar criteris d'anotació als annotadors de la campanya.

3.2 Executant l'extractor

L'extractor es pot executar sobre una canço del projecte des del propi Annotator. Pot ser qualsevol programa de línia de comandes que, donat un fitxer d'àudio, n'extregui la informació en el format XML que entén l'Annotator. L'Annotator es distribueix amb dos exemples de programes extractors: Un que extreu descriptors instantanis de baix nivell i un altre, que es comenta més avall, que n'extreu acords. El programa d'extracció també ha d'acceptar opcions de línia de comandes per generar l'esquema de dades que produeix.

La idea original era fer extractors endollables com a llibreries dinàmiques. Això feia més complicada, però, la integració d'extractors existents. Fent la integració de extractors a nivell d'executables ens ha permés integrar extractors ja existents simplement amb un envolcall Python, i fer servir extractors sense que sigui molt obstacle la seva situació de propietat intel·lectual, sovint complicada en la convivència de partners acadèmics i industrials. La granularitat dels extractors tampoc és problema. Els formats XML que fem servir per definir l'esquema i per contenir les dades fan molt fàcil barrejar informació de diverses fonts o filtrar-la.

3.3 Validant l'extracció

La forma més directa de validar les anotacions és la visual. L'Annotator ofereix vistes aliniades horitzontalment en el temps, anomenades *vistes d'evolució*. A les *vistes d'evolució* podem veure l'evolució dels atributs numèrics instantanis o la posició, i duració, de les divisions en segments.

També es poden visualitzar (i editar, si cal) els valors dels atributs que hi ha tant a nivell de cançó, com a nivell de segment quan en seleccionem un. Cada atribut s'edita de forma diferent depenent del seu tipus definit a l'esquema.

Les anomenades *vistes instantànies*, són un altre grup de vistes, que són molt útils per a la validació. Permeten visualitzar els atributs en un punt temporal concret. Aquestes vistes es poden sincronitzar amb la reproducció de l'àudio, oferint una correspondència directa del que es veu amb el que sona.

En aquest sentit, l'anotador ofereix una via adicional de contrastació, les *auralitzacions*. Una auralització es un só de referència que es sintetitza a partir de la informació extreta o anotada. Aquest so es reproduïeix sincronitzadament amb la música original, serveix per validar l'anotació. Per exemple, l'Annotator ve amb dos auralitzacions una que fa un click amb l'inici dels segments i una altre que sintetitza un oscilador amb la freqüència controlada pel valor de un atribut instantani de tipus real.

Tots els elements citats, vistes d'evolució, editors de valors d'atributs, vistes instantànies i auralitzacions, són ampliables a d'altres tipus mitjançant plugins. Cada vista pot esbrinar si un atribut té un tipus al qual es pot aplicar o no. Depenent d'això es dona o no la possibilitat a l'usuari de fer servir aquesta vista.

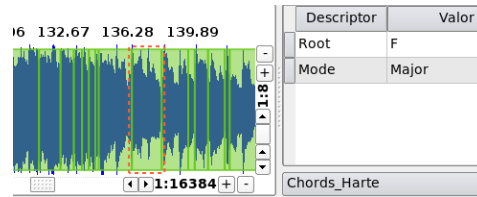


Figura 4: Quin acord ha escollit l'extractor per aquest segment?

4 Un cas concret, els acords

4.1 L'extractor d'acords

L'extractor d'acords que es distribueix amb l'Annotator ha estat desenvolupat com a fruit d'una col.laboració amb el Departament d'Electrònica de la Queen Mary University of London. Està basat en un algorisme de Chris Harte [6] amb algunes optimitzacions i millores.

L'algorisme original de Chris Harte és el següent: Per cada instant es fa una transformada de Fourier d'alta resolució; es mapeja la transformada d'una escala lineal a una de logarítmica de tal manera que quedin 3 *bins* per semitò⁸; es sumen tots els bins per octaves; es cerquen i s'interpolen els pics; es troba l'afinació més probable dels pics al llarg de la canço; es corregeix la desviació de l'afinació de referència; s'atenuen els pics que estan fora de to; s'obté, per a cada instant, un histograma amb la preponderància de cada nota, anomenat PCP (Pitch Class Profile); es suavitzen els PCP's amb els valors dels PCP's anteriors, es correla el PCP amb un seguit de patrons PCP d'acords ideals; es suavitzen les correlacions, i el patró que té més puntuació és el que es selecciona.

L'algorisme original es va portar de Matlab a C++ i es van fer optimitzacions i algunes millores:

- l'afinació es fa adaptativa per cada frame tal que es pot fer servir en streaming
- s'inclou un acord **None** amb tots els pesos iguals per totes les notes que serveix per tenir un llindar a partir del qual un acord detectat no es considera acord.
- es va substituir el suavitzat de les correlacions per una funció simbòlica que prima els acords llargs i neteja els puntuals.

4.2 Les dades extretes

La principal dada extreta són els segments d'acord amb els atributs **Root** (Do, Re...) i **Mode** (Major, Menor, Dominant, Disminuït...). Però de cara a validar l'algoritme també s'ofereixen altres dades com ara l'evolució del punt d'afinació en cada instant, les distàncies relatives del primer i del segon acord candidat al acord **None** i entre ells en cada instant, el PCP abans de suavitzar i les correlacions dels acords.

⁸Les distàncies en hertzs entre semitons son proporcionals a la freqüència.

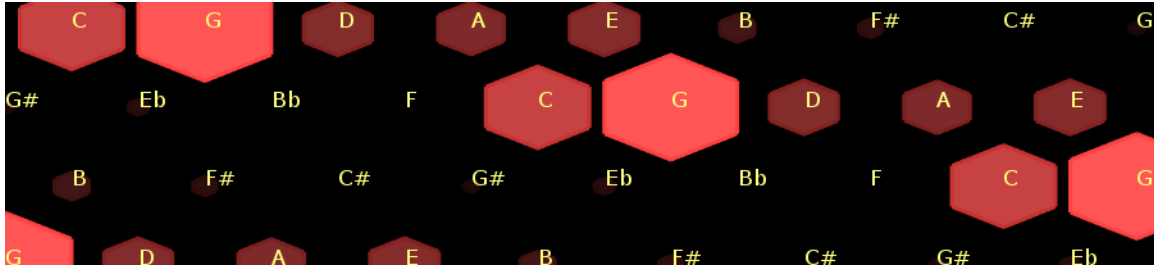


Figura 5: Vista Tonnetz

4.3 Visualitzacions especials

Per visualitzar aquestes dades millor s'han afegit a l'Annotator un parell de vistes instantànies idònies per a l'extracció de acords: El *Tonnetz* i el *KeySpace*

Tonnetz, vol dir en alemany 'xarxa de tons'. Consisteix en disposar els tons, el PCP, com si fossin hexàgons d'un rusc en una superfície toroidal tal que el tons adjacents en horitzontal tenen una distància de 7 semitons (una quinta). els adj en direcció sudoest-nordest tenen una distància de 4 semitons (una terça major), i els que són adjacents en direcció nordoest-sudest tenen una distància de 3 semitons (una terça menor).

L'avantatge d'aquesta disposició és que els tons que formen els acords típics estan junts i dependent del mode (major, menor, septima dominant, augmentat, disminuït...) conformen figures identificables. Per exemple, els acords majors es disposen com a triangles amb una punta cap adalt, mentres que els acords menors es disposen com a triangles amb una punta cap avall. La figura 5, per exemple, mostra un do major (C-E-G).

L'altra visualització instantània és el KeySpace (figura 6) que visualitza la correlació dels acords però, només dels majors i menors. Aquesta vista va ser ideada per Emila Gómez i Jordi Bonada [4] i el que representa és també la probabilitat de cada acord però, disposats ara per afinitat harmònica. Això vol dir que els acords que comparteixen notes estan propers. D'aquesta manera sempre veiem un acord preponderant rodejat d'una taca compacta dels acords que comparteixen 2 o, més lluny i enmortits, 1 nota. Si no estiguessin ordenats així es veuria munt de taques disperses en l'espai, i seria molt difícil discernir quina és la més intensa.

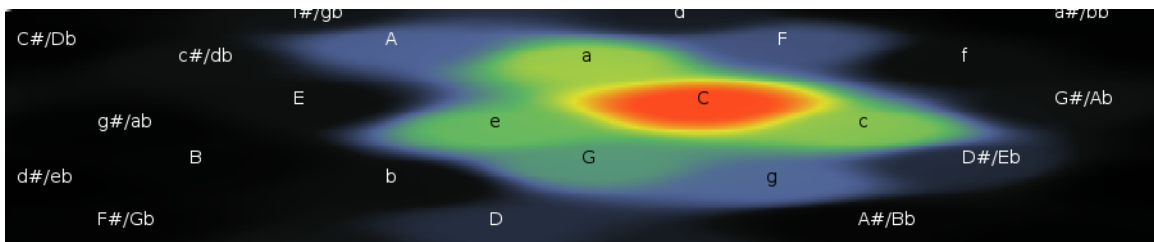


Figura 6: Vista KeySpace

5 Aspectes tecnològics interessants

5.1 SCons

Durant el desenvolupament de l'aplicació hem migrat el sistema de construcció, d'un sistema propi que complementava l'eina *make* cercant dependències a SCons. L'impressió que ens ha donat ha estat realment molt positiva. S'ha agilitzat el procés d'afegir llibreries i binaris diferents. També ha ajudat molt el fet que tinguem un distribució binaria de CLAM.

5.2 Qt 4

L'aplicació inicialment estava desenvolupada amb Qt 3. Vam decidir migrar a Qt 4 un cop l'aplicació estava ja molt avançada. La migració és un procés laboriós, però, sense massa entrebancs importants que té com a recompensa un codi generalment més clar i simple. Qt 4 aporta eines de migració que, tot i que no ho migren tot automàticament, fan molta feina bruta.

A Qt 4, s'han fet moltes millores. La compilació dels arxius d'interfície (.ui) s'ha simplificat molt. La gestió dels recursos (principalment les imatges) és molt més elegant. En general l'API ha fet canvis molt encertats que simplifiquen i racionalitzen l'ús. L'únic, potser, una mica massa complex es la nova API per vistes de taula, llista i arbre. S'introdueix el model vista controlador[9], que fet servir en un codi des de zero es prou bo, però en portar codi existent, que com les vistes de l'Annotator, ja implementa un model vista controlador, més adequat al problema, resulta un inconvenient.

Un resultat colateral d'aquesta migració ha estat el desenvolupament d'una *tool* d'SCons⁹ per compilar programes en Qt 4. Abans d'això només existia una tool per a Qt 3 que no era pas adient per a Qt 3. Aquesta tool serà probablement integrada a la distribució oficial d'SCons.

5.3 Python

Una altra tecnologia que hem introduït durant el desenvolupament és Python. Tot i que els desenvolupadors no teniem tant domini del llenguatge Python i les seves llibreries, com ho tenim de C++, ens ha sorprès gratament la agilitat i el ritme de desenvolupament que es pot assolir amb Python.

Python s'ha fet servir en diversos aspectes del desenvolupament: els programes d'agregació de fonts i conversió de dades, els adaptadors de programes extractors externs, el client d'accés BOCA, BOCA mateix fent servir `mod_python` i el framework Django, els serveis web que es fan servir per muntar el projecte a partir d'una tasca BOCA.

Ara bé, fer servir python ens obligava a instal·lar Python i les llibreries dependents, la qual cosa és un inconvenient a Windows. Per sort hem trobat l'aplicació Py2exe que genera, no sense problemes, un binari que no necessita un python instal·lat.

⁹<http://www.iaa.upf.edu/~dgarcia/Coddery/sconstools.html>

6 El futur

Un cop finalitzat el projecte SIMAC, el desenvolupament de l'Annotator recau plenament en el projecte CLAM. Una de les primeres coses que cal adreçar és identificar què de l'Annotator és d'ús genèric i cal integrar com a part de CLAM. Algunes coses que caldria integrar serien la definició d'esquemes, alguns tipus bàsics i les visualitzacions. També molt treball que s'ha fet amb l'Annotator en quant el port a Qt4, el sistema de build i l'empaquetat cal projectar-ho a la resta d'aplicacions de CLAM.

En quant a l'aplicació en sí, són necessàries millores en la interacció amb l'usuari tenint molt en compte l'ergonomia. Per exemple, donada la repetitivitat de les tasques d'anotació és molt urgent dotar-lo de drecceres de teclat prou potents com per poder fer servir el ratolí el mínim. A vegades el ratolí no té massa precisió per algunes tasques, com el posicionat del segments i unes drecceres ben plantejades poden ser de molt ajut.

També relacionades amb l'ergonomia calen funcionalitats com una funció *Desfer* i poder aplicar canvis en els atributs a tot un conjunt d'objectes (cançons, segments...). Un mode de funcionament que ens ha agradat molt i que ens agradaria fer servir es el que fa servir el programa de etiquetació d'mp3 EasyTag. EasyTag permet canviar un valor d'un atribut de cop a un conjunt de fitxers o aplicar-ne transformacions típiques.

La tercera línia de futur relacionada amb l'Annotator té a veure amb els endollables. Per un costat cal definir i estabilitzar les interfícies dels endollables. Per un altre costat, cal definir-ne més endollables per l'aurallització, vistes d'evolució, vistes instantànies i editors d'atributs.

Algunes vistes que seria interessant de tenir són vistes d'evolució d'arrays numèriques que permetin veure, per exemple, l'espectrograma, el cromograma o els PCP's. També vistes instantànies que permeting veure amb detall instàncies dels anteriors en un instant del temps. Les versions polars són molt útils per els PCP's i els cromogrames per destacar el fet de que el primer i l'últim bins són contigus, cosa que en la visualització cartesiana no queda tan clara. No costarà gaire incorporar aquestes vistes donat que versions cartesianes i polars ja existeixen tant a CLAM com a l'aplicació lliure Sonic Visualizer, també desenvolupada en el context del projecte SIMAC i alliberada recentment. De fet, el Sonic Visualizer té algunes vistes molt maques i és molt probable l'intercanvi de codi entre ambdúes aplicacions.

7 Agraïments

Els autors de l'article volen agrair el treball dels altres desenvolupadors que han intervingut en el desenvolupament de l'aplicació, Ismael Mosquera, Jordi Massaguer, Xavier Oliver, i en general l'equip de CLAM.

També volem agrair les aportacions de Chris Harte i Kate Noland en l'extractor d'acords i d'Emilia Gómez i Jordi Bonada per la vista *KeySpace*.

El desenvolupament del CLAM Music Annotator ha estat parcialment finançat pel projecte europeu SIMAC (IST-507142). El projecte CLAM està parcialment finançat per la Generalitat de Catalunya (expedient 200/05 ST).

Referències

- [1] AMATRIAIN, X., ARUMÍ, P., AND RAMÍREZ, M. CLAM, Yet Another Library for Audio and Music Processing? In *Proceedings of the 2002 Conference on Object Oriented Programming, Systems and Application (OOPSLA 2002)(Companion Material)* (Seattle, USA, 2002), ACM.
- [2] AMATRIAIN, X., MASSAGUER, J., GARCIA, D., AND MOSQUERA, I. The clam annotator: A cross-platform audio descriptors editing tool. In *Proceedings of 6th International Conference on Music Information Retrieval* (London, UK, 2005).
- [3] CELMA, O., RAMÍREZ, M., AND HERRERA, P. Foafing the music: A music recommendation system based on rss feeds and user preferences. In *Proceedings of 6th International Conference on Music Information Retrieval* (London, UK, 2005).
- [4] GÓMEZ, E., AND BONADA, J. Tonality visualization of polyphonic audio. In *Proceedings of International Computer Music Conference 2005* (Barcelona, 2005).
- [5] GOUYON, FABIEN; WACK, N., AND DIXON, S. An open source tool for semi-automatic rhythmic segmentation. In *Proceedings of 5th International Conference on Music Information Retrieval* (Barcelona, October 2004).
- [6] HARTE, CHRISTOPHER; SANDLER, M. Automatic chord identification using a quantised chromagram. In *Proceedings of 118 Convention of Audio Engineering Society* (Barcelona, May 2005).
- [7] HERRERA, P., BELLO, J., WIDMER, G., SANDLER, M., CELMA, O., VIGNOLI, F., PAMPALK, E., CANO, P., PAUWS, S., AND SERRA, X. Simac: Semantic interaction with music audio contents. In *Proceedings of 2nd European Workshop on the Integration of Knowledge, Semantic and Digital Media Technologies* (Savoy Place, London, UK, 2005).
- [8] JENSEN, K. E. A. Rhythm-based segmentation of popular chinese music. In *Proceedings of 6th International Conference on Music Information Retrieval* (London, 2005).
- [9] KRASNER, G. E., AND POPE, S. T. A cookbooc for using the model-view controller user interface paradigm in smalltalk-80. *Journal of Object-Oriented Programming* (1988).
- [10] SJLANDER, K., AND BESKOW, J. Wavesurfer - an open source speech tool, 2000.