

Rosario, Argentina, 12 y 19 al 23 de mayo de 2014

Computación Musical

Herramientas informáticas actuales de código abierto

Luis Tamagnini^a

^a Universidad Nacional de Rosario

Resumen

El siguiente texto analiza algunos aspectos generales de la Computación Musical. Se expone una descripción conceptual general y una breve reseña de su historia. A continuación se refiere más específicamente a algunos lenguajes de programación de código abierto y otras herramientas informáticas actuales desarrolladas para la práctica de la misma.

1 Acerca de la Computación Musical

Desde los primeros años de existencia de las ciencias de la computación, los científicos e investigadores que promovieron su enorme desarrollo hacia mediados del S. XX, experimentaron en la utilización de los (en ese entonces nuevos) recursos computacionales para la creación artística; no sólo de música y sonidos, sino también de literatura y artes visuales o gráficas. Estas investigaciones se llevaron a cabo dentro de un marco más global de la informática y la ingeniería electrónica; la Inteligencia Artificial.

La Computación Musical es una disciplina en la cual se utilizan tecnologías informáticas para componer música. Esta comprende el desarrollo y la aplicación de nuevas técnicas computacionales en aspectos básicos de la praxis musical como la síntesis de sonido, el procesamiento de audio digital, la acústica y psicoacústica, entre otros. Los orígenes de la Computación Musical pueden ser rastreados en los inicios mismos de la música electrónica y en las primeras experiencias e innovaciones con instrumentos electrónicos realizadas a principios del S. XX.¹

En la actualidad, en virtud de la enorme expansión de los medios digitales que tuvo lugar en los últimos 20 años, y ante una realidad en la cual es difícil encontrar algún producto musical que no haya pasado en algún momento por una computadora, podemos corregir o “afinar” un poco la definición considerando que: una pieza de computación musical es un “objeto musical que no puede haber sido creado sin la utilización de una computadora”.

2 Breve reseña histórica

Durante los años 50 se desarrollan dos nuevas modalidades tecnológicas que impulsan enormemente la computación musical; la síntesis digital de sonido y los programas de composición algorítmica. En 1957, Max Mathews, desarrolla en los laboratorios Bell el primer

lenguaje de computación musical ampliamente utilizado y difundido, MUSIC I.² Este lenguaje fue muy influyente y con él se estableció un modelo de diseño para este tipo de software, que aún en la actualidad es utilizado en la mayoría de los lenguajes contemporáneos de CM.³ Estos logros fueron presentados en la revista Science en 1963 generando gran interés. La IBM 704, una computadora construida en 1954 y para la cual Mathews escribió MUSIC I era por aquel tiempo el hardware más avanzado. Ésta fue, además, la primera computadora producida en masa capaz de operar con aritmética de punto flotante, una condición sine qua non para resolver la síntesis de audio digital (Figuras 1 y 2).



Figura 1. IBM 704 (1954)

En cuanto a la utilización de algoritmos para estructurar el discurso musical, podemos mencionar a la Suite Illiac, como la primer pieza musical completamente compuesta por una computadora.⁴ Esta pieza fue creada en 1954 por Lejaren Hiller y Leonard Isaacson, quienes programaron la máquina para generar la partitura. Ésta fue luego interpretada por un cuarteto de cuerdas. La pieza se compone de cuatro movimientos llamados “experimentos” en los que se utilizan diferentes algoritmos. En el primero se explora la generación de “cantus firmus”, en el segundo se generan segmentos a cuatro voces, basados en reglas, el tercero juega con el ritmo y la dinámica según diferentes instrucciones de ejecución. En el cuarto “experimento” se utilizan varios modelos de probabilidad, gramática generativa y cadenas de Markov.

Otro de los compositores pioneros en la utilización de computadoras para escribir música fue Iannis Xenakis (1922 - 2001). Él escribió programas en lenguaje FORTRAN para generar información numérica que luego transcribía a notación musical. Estas partituras eran luego interpretadas por conjuntos instrumentales tradicionales. Entre 1956 y 1962 Xenakis escribió una serie de piezas llamadas ST; ST/4 para cuarteto de cuerdas, ST/10 para conjunto de 10 instrumentistas y ST/48 para orquesta. Si bien estas partituras podrían haber sido escritas a mano, la alta demanda de cálculo que Xenakis utiliza para mapear matemática de probabilidad a notas musicales motivó el uso de la computadora.⁵

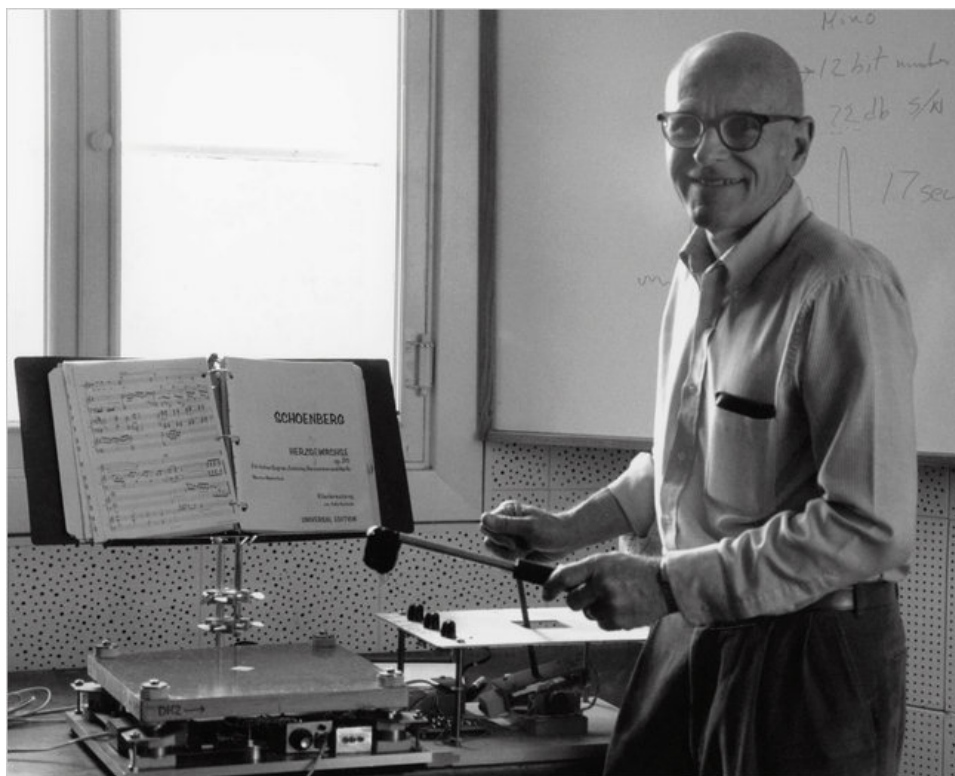


Figura 2. Max Mathews.

3 Descripción conceptual

Miller Puckette explica en el texto “Computing While Composing” que el campo de estudios de la Computación Musical puede dividirse en dos áreas o ramas fundamentales: la primera concierne a la manipulación de sonidos musicales, y la otra, a la representación simbólica de la música.⁶ Estos dos campos se ponen de manifiesto en el lenguaje MUSIC I, de Max Mathews, por un lado; y en la Suite Illiac, de Lejaren Hiller, por el otro.

En la actualidad, se reconoce a estas sub áreas como: Música Generada por Computadora (CGM) y Composición Asistida por Computadora (CAC).⁷

La CGM y la CAC han experimentado desde su aparición un desarrollo, al mismo tiempo paralelo y relacionado, y por otro lado diferenciado desde un punto de vista geográfico y técnico. En el caso de la CGM, se destaca a partir de los años 80, la posibilidad de correr sus algoritmos en tiempo real, lo cual ha resignificado la cuestión notablemente. Por el lado de la CAC, el aumento del potencial computacional experimentado por las máquinas en los últimos 30 años ha derivado en una marcada complejización de los algoritmos utilizados para la misma, que pasaron del cálculo de unos cientos a varios millones de operaciones aritméticas por nota.

4 El software de computación musical⁸

Hoy en día existe una tendencia en los compositores a utilizar no un único lenguaje de computación musical sino una serie de ellos. Esto era antes algo complicado, sobre todo debido a la dificultad para desarrollar destreza programando en antiguos lenguajes, algo que podía tardar años. Sin embargo, a medida que los lenguajes fueron siendo más abiertos y modulares, la posibilidad de intercambio de información entre ellos se ha incrementado. Al mismo tiempo también resulta en la actualidad más sencillo y menos costoso en tiempo aprender a programar, sobre todo en lenguajes gráficos como Pure Data u Open Music por ejemplo. Muchos y muy diferentes lenguajes existen hoy que pueden incluso ensamblarse entre si en una red computacional compleja.⁹

En relación a esto, quizás como un caso de esa tendencia hacia la interoperabilidad, está el cada vez más extendido uso de la CAC en la manipulación directa de los sonidos (y no a

través de la mediación de una partitura). Este paradigma implica la resolución simultánea de la CGM y la CAC; y sugiere que en un futuro la distinción entre ambas pueda simplemente desaparecer.¹⁰ Esto contrasta enormemente con los inicios de la Computación Musical en donde el resultado de la ejecución de un programa era una pila de tarjetas perforadas, o con lo que sucedía hace sólo 10 ó 15 años en donde los compositores necesitaban un hardware MIDI para poder oír sus ideas musicales. Uno de los lenguajes más recientes y que puede satisfacer esta demanda es SuperCollider.¹¹

5 SuperCollider

SuperCollider es lenguaje de programación musical originalmente publicado en 1996 por James McCartney, para síntesis de sonido en tiempo real y composición algorítmica. Desde entonces ha evolucionado a un sistema desarrollado tanto por científicos como por artistas sonoros. Es un lenguaje dinámico, eficiente y expresivo y provee un marco para la investigación acústica, la música algorítmica y la programación interactiva. Publicado bajo los términos de la Licencia Pública General GNU en 2002, SuperCollider es software libre. La versión más reciente es la 3.6.5 publicada en abril de 2013.

El entorno de SuperCollider está dividido en dos componentes, un servidor llamado *scsynth*, y un cliente denominado *sclang*. Estos componentes se comunican entre sí mediante el protocolo OSC (Open Sound Control).

El lenguaje combina la estructura orientada a objetos de Smalltalk y características de lenguajes funcionales con una sintaxis similar a C.

El servidor soporta una interfaz de programación simple en C que hace fácil escribir algoritmos de sonido eficientes. Estos reciben el nombre de *Ugens* (o *unit generators*). Debido además a que el servidor es controlado en forma remota vía OSC, también es posible utilizarlo con otros clientes, que pueden ser otros lenguajes u otras aplicaciones.

Otras de las poderosas características del servidor son el soporte para cualquier número de canales de entrada y salida, incluyendo situaciones multicanal masivas; acceso a una estructura ordenada de nodos de síntesis que define en forma precisa el orden de ejecución; un sistema de buses que permite reestructurar dinámicamente el flujo de la señal; buffers para escritura y lectura y cálculo a diferentes tasas dependiendo de las necesidades, tasa de audio, tasa de control y tasa de demanda.

SuperCollider corre sobre sistemas Mac OS X, Linux y Windows, entre otros y para cada uno de ellos existen un número de entornos de desarrollo y clientes que pueden utilizarse. Ha sido demostrado además que puede correr en sistemas iOS y Android.

Por el lado del cliente, este incluye un intérprete para el lenguaje, el cual permite al usuario escribir código en dicho lenguaje que el mismo “interpreta” generando comandos para controlar al servidor de audio. En particular los comandos OSC y el propio servidor comparten una perspectiva de bajo nivel y puede resultar complejo escribir código a estas instancias. Pero por otro lado el lenguaje propiamente dicho es de alto nivel, totalmente orientado a objetos y permite al usuario lograr mayor expresividad desde su código. Típicamente, el intérprete traduce el código escrito en lenguaje SuperCollider a mensajes OSC para el servidor de audio¹² (Figuras 3 y 4).

6 Open Sound Control¹⁵

Open Sound Control (OSC) es un protocolo de comunicación entre computadoras, sintetizadores y otros dispositivos multimedia, que está optimizado para los sistemas de redes modernos. Brindando los beneficios de las redes actuales al mundo de los instrumentos musicales electrónicos, las ventajas de OSC incluyen; interoperabilidad, mayor precisión y flexibilidad, y una documentación mejorada.

Este simple, aunque potente protocolo provee todo lo necesario para el control en tiempo real del sonido y otros procesos multimediales al tiempo que se mantiene flexible y fácil de implementar.

OSC fue y continúa siendo desarrollado en el Center for New Music and Audio Technology (CNMAT), en la Universidad de Berkeley.¹⁶

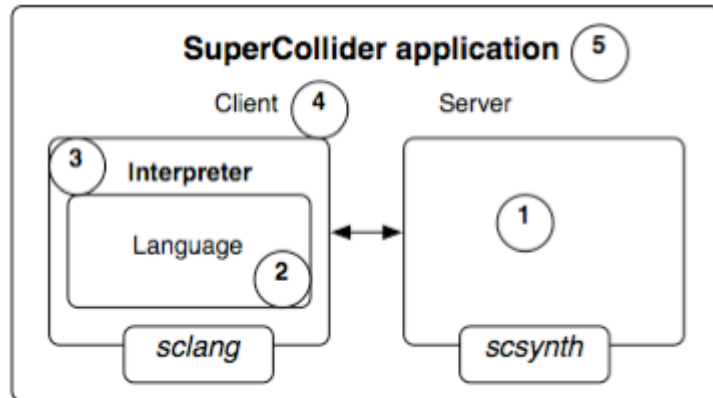


Figura 3. Estructura de SuperCollider.¹³

```
// UNA FUNCION GENERADORA DE AUDIO
(
play({CombN.ar(
  SinOsc.ar(midicps(
    LFNoise1.ar(3, 24, LFSaw.ar(
      [5, 5.123], 0, 3, 80))), 0, 0.4), 1, 0.3, 20)*0.1})
)
```

Figura 4. Ejemplo del código de SuperCollider.¹⁴

Como se ha dicho anteriormente, cualquier cliente que implemente Open Sound Control puede ser utilizado para controlar el servidor de SuperCollider, o para recibir datos desde SuperCollider. En la última década, se ha dado una tendencia muy fuerte al desarrollo de entornos o frameworks para la programación creativa, fundamentalmente computación gráfica y multimedial. Entre ellos se destacan Processing,¹⁷ un framework basado en Java y OpenFrameworks,¹⁸ basado en C++, entre otros.

Un framework, o marco de trabajo, es un paquete de soluciones pre-ensambladas orientadas a la programación en áreas específicas.¹⁹ En el caso de Processing, desarrollado en el Media Lab del MIT y publicado inicialmente en 2001, además se trata de una herramienta pedagógica, en función de una sintaxis simplificada y de proveer además un IDE o entorno de desarrollo muy mínimo y fácil de utilizar (Figura 5).

7 A/V Performance electrónica audiovisual

En mi trabajo como artista utilizo software libre para la generación de audio, video, imágenes digitales y sistemas interactivos, entre otras cosas. En A/V, una performance electrónica audiovisual que voy a interpretar en el concierto que sigue a esta conferencia, tanto el audio como las visuales son generados en tiempo real. Para la generación del audio utilizo SuperCollider, para resolver aspectos tanto de la CGM como de la CAC; mientras que las imágenes digitales son generadas en Processing, con el soporte de opengl²⁰ para optimizar el rendimiento. En este caso la componente visual es reactiva a la componente sonora. El control en vivo de la interpretación se realiza mediante un controlador MIDI standard.²¹

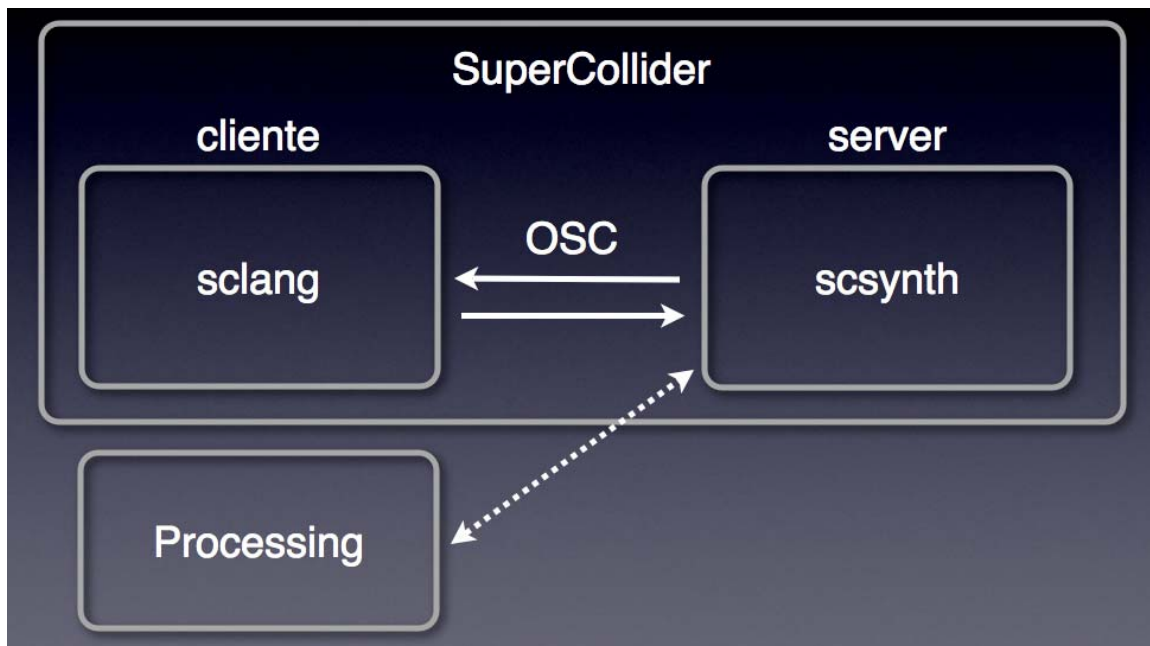


Figura 5. Control de SuperCollider desde cliente externo.

Referencias

- ¹ http://en.wikipedia.org/wiki/Computer_music
- ² <http://en.wikipedia.org/wiki/MUSIC-N>
- ³ http://en.wikipedia.org/wiki/Unit_generator
- ⁴ <https://www.youtube.com/watch?v=n0njBFLQSk8>
- ⁵ <https://www.youtube.com/watch?v=9KsWpuqTYLU>
- ⁶ Puckette, M. 2006. "Computing While Composing." Preface to the OM Composer's book (Carlos Agon *et al.*, eds.), Editions Delatour France / IRCAM.
- ⁷ Ibid.
- ⁸ http://en.wikipedia.org/wiki/Audio_programming_language
- ⁹ Puckette, M. 2006. "Computing While Composing." Preface to the OM Composer's book (Carlos Agon *et al.*, eds.), Editions Delatour France / IRCAM.
- ¹⁰ Ibid.
- ¹¹ <http://supercollider.sourceforge.net/>
- ¹² <http://en.wikipedia.org/wiki/SuperCollider>
- ¹³ <http://doc.sccode.org/Guides/ClientVsServer.html>
- ¹⁴ Wilson, S., Cottle, D. and Collins, N. (eds). 2011. The SuperCollider Book. Cambridge, MA: MIT Press.
- ¹⁵ <http://opensoundcontrol.org/>
- ¹⁶ http://en.wikipedia.org/wiki/Open_Sound_Control
- ¹⁷ <https://www.processing.org/>
- ¹⁸ <http://www.openframeworks.cc/>
- ¹⁹ <http://es.wikipedia.org/wiki/Framework>
- ²⁰ <http://es.wikipedia.org/wiki/OpenGL>
- ²¹ Registro en video del concierto: https://www.youtube.com/watch?v=c_M3wOWSaQ