Virtual Composer: a tool for musical composition

Riccardo Santoboni
Bari Conservatory of Music
r.santoboni@mclink.it

Mauro Clementi
MARR software
pigniaschi@tiscalinet.it

Anna Rita Ticari
Accademia Musicale Pescarese
MD9077@mclink.it

http://www.thevirtualcomposer.com

Abstract

The idea for Virtual composer (VC) grew up from the master course of composition held by Santoboni since 1998. Main aim of this software is composing assistant: composer chose architecture of composition, language styles, musical system and “engines” for generation and “modelling” of pitches, duration and timbre. The approach is an “object orientated” approach. In this way the engines can both produced objects than modify the properties of objects themselves. Clementi developed the software while Ticari mainly prepared mathematical and stochastic engines.

1. Introduction

A musical composition can be thought as a structure in which its elements are more or less related each other. In any case it is possible to represent it, as a hierarchic tree with the object “Composition” at the top and the object “Note” at the bottom. The object oriented approach for the composition helped us to a better definition of the elements of a composition, to focus the structure of a musical composition and to manipulate the proprieties of the objects of composition with deterministic and stochastical engines. A similar approach can also be used to analyse musical works.

2. Architecture

In our approach the objects of a composition are, in hierarchical order: Composition, Track, Section, Measure, Note. Although the sections and measures approach may seem a classical approach, the result is flexible to set both historicized music than contemporary music. Composition properties are, among others: Author name, title, number of tracks, measures, filename, etc. Track properties are polyphony, midi channel or synthesis technique, kind of event graphic representation (staff, segments, piano roll, etc.). Section are mainly time windows, in which we can chose the engines for the generation of “sound events”.

In each section engines generate the allowed pitches and durations; it is possible to chose some of them by the same (recursion) or others engines. Measure properties are mainly for historicized music and include metric patterns, accents, etc. Each section can be homometric, multimetric or polymetric. Note properties include information for the overlapping of different note in tracks (consonance-dissonance), for articulations in historicized music, and in generally include arbitrary settings of rules both for harmony than counterpoint.

Engines

Virtual composer is an open structure where insert new engines. Engines can be catalogue in two sets: generation engines and manipulation engines. Typical generation engines are deterministic engine, based on mathematical successions, progressions, logical criteria and stochastical engines based on memory and with no memory processes. Generation engines produce the section setting, that means allowed frequencies (pitches) and durations for each section. Among the generation engines, a new approach is the geometrical approach represented by the polygonal engine: a circle represents a unit: an octave (or other chosen interval) for pitch as well as a measure (or a time window) for duration. With an arbitrary inscribed polygon we chose the allowed pitches (for scales) and duration patterns. Rotation is a Polygon engine property: It is possible to use several polygons together, with different rotation to generate complex rhythm and scales. In this way it is easy to build isorhythmic structures as G. De Machaut motets or complex polymetric sections based i.e. on Messian modes, or simple strophic songs. Among the manipulation engines we have the Profile engine for melodic contours, Chord engine for harmony, ect. Fuzzy and genetic algorithms for the engines are now in testing as well as fractal algo for composing architecture.

4. Saving the result

VC 2.0, uses a database for data storing. XML format for easier data managing will be implemented in next release.