# "PWGL: A SCORE EDITOR FOR CSOUND"

Massimo Avantaggiato
"G.Verdi" Conservatoire – Milan –
Italy

mavantag@yahoo.it

## Introduction

PWGL[1] has stood out, since its introduction in Electronic Music classes, as an important tool for the development of the musical creativity of students at the "G.Verdi" Conservatoire.
The uses of PWGL are manifold: other than being a great teaching aid for rebuilding the composition of historical pieces, it offers new opportunities in the creation of original pieces and the application of several techniques of sound synthesis. I've concentrated on the need to create functional links between PWGL and Csound, in order to integrate the processes of algorithmic composition with those of sound synthesis.

A piece is usually written with using thousands of lines of instructions. However, most musicians do not compose pieces writing note by note, but use programmes to produce scores.
Such programmes, known as score generators [2], aim at relieving the composer of the repetitive task of typing long lines of code.

PWGL, a programme of algorithmic composition based on Common Lisp [3], can be employed as a score generator and interfaced with Csound, a widespread sound renderer based on the C language; it can also become a powerful driving force of synthesis [4].
PWGL enables an unskilled programmer, without the support of external libraries to do the following:
- produce electronic gestures with full control of the sound parameters involved in the sound-"composing" process: instrument, attack, note, amplitude, frequency, pan, reverb, delay etc.;
- make a quick rendering in Csound, to repeat processes and correct PWGL patches; implement , by using simple algorithms, several techniques of synthesis.

## 1) ADDITIVE SYNTHESIS AND KARPLUS

Firstly, take for instance the synthesis of Karplus and Strong: the patch I'm going to show allows us to choose among spectra which will undergo a process of acceleration or deceleration.
The resulting gesture will be synthesized in Csound, by means of the following opcode [5]:

```
ar pluck kamp, kcps, icps, ifn, imeth[iparm1,iparm2]   [6]
```

1 "PWGL could be seen as an attempt to fill the gap between several different aspects of music tuition. It is our belief that PWGL could be established as a pedagogical tool for academia"[7].
2 Score Generators have been written in several languages: C, C++, Java, Lisp, Lua and Python. In one of his writings Goggins makes a list of different types of Score Generators and Composing Environments, including AthenaCl, Blue, Common Music, Pure Data [4].
3 For an in-depth study see: [1,21].
4 It's an option for the use of PWGLSynth, which PWGL is already equipped with[8-10].
5 See references for an in-depth study of Karplus and Strong's synthesis techniques [5, 6].

First of all, the instrument is written:

```
instr 1
idur = p3
iamp = p4
ifrq   = p5
ipanl  = sqrt (p6)
ipanr = sqrt (1-p6)
kenv linseg 1, p3*.50, 1, p3*.25, 0
a1 pluck iamp, ifreq, ifreq, 0, 6
outs a1*kenv* ipanl, a1*kenv* ipanr
endin
```

Secondly, the patch is produced (see figure 2):

- in 1) the starting sequence of notes is singled out;
- in 2) the abstract- box is made. Its function is to produce the data for the synthesis and, if necessary, go on with the acceleration or deceleration of the material;
- in 3) the data is collected in a text-box to implement the Csound synthesis.

The abstract-box, seen from the outside, consists of a series of sliders linked with the inputs (see figure 1-2). Each parameter-field can be matched by one or more inputs (see figure 3):

**1) INSTRUMENT (p1):** the values given to the instruments are repeated as many times as the frequencies composing the spectrum.
**2) CREATION TIME (p2):** the attacks of the original sequence are rescaled or not, by means of a parabolic function, in order to bring about the acceleration or deceleration of the starting material;
**3) LENGTH OF NOTES (p3):** the length of notes, ranging between a minimum and maximum value, is defined by using g-scaling and taking into account the distance between the creation times previously defined;
**4) AMPLITUDE (p4):** it can be expressed in absolute values or be defined in Db;
**5) FREQUENCIES (p5):** they are the frequencies of the starting spectra; if midi values are employed it's necessary to convert them by using the function "Patchwork Midi to Frequencies";
**6) STEREO PANNING** [7] **(p6):** thanks to a parabolic function it ranges from 1 (all the sound is on the left) to 0 (all the sound is on the right).
As for the parameters concerning creation time, length, amplitude and pan, the type of interpolation applied is based on a concave/convex curve, even if it's not the only type of interpolation available.

---

6 - kamp = sound amplitude;
- kcps = target frequency.
- icps = value (hertz) setting out the length of the table.
It usually equals kcps, but can be increased or decreased to get special timbre effects.
- ifn = number of the function to initialize the table.
When ifn = 0, the table is filled with a random sequence of values (original algorithm of Karplus and Strong)
-imeth = method used to change the values of the table during the sound-producing process. There are six of them [3].

[7] "The most popular spatial illusions are horizontal panning – lateral sound movement from speaker to speaker – and reverberation – adding a dense and diffuse pattern of echoes to a sound to situate it in a larger space.
Vertical panning (up and down and overhead) can also create striking effects in electronic music [19]."
Mickelson mentions strategies for panning and offers a grading of some basic strategies: Simple Pan; Square Root Pan; Sine Pan; Equal Power Pan; Delay and Filtered Panning [15]


**Figure 1:** The sliders of the abstract-box employed to transfer data to Csound.

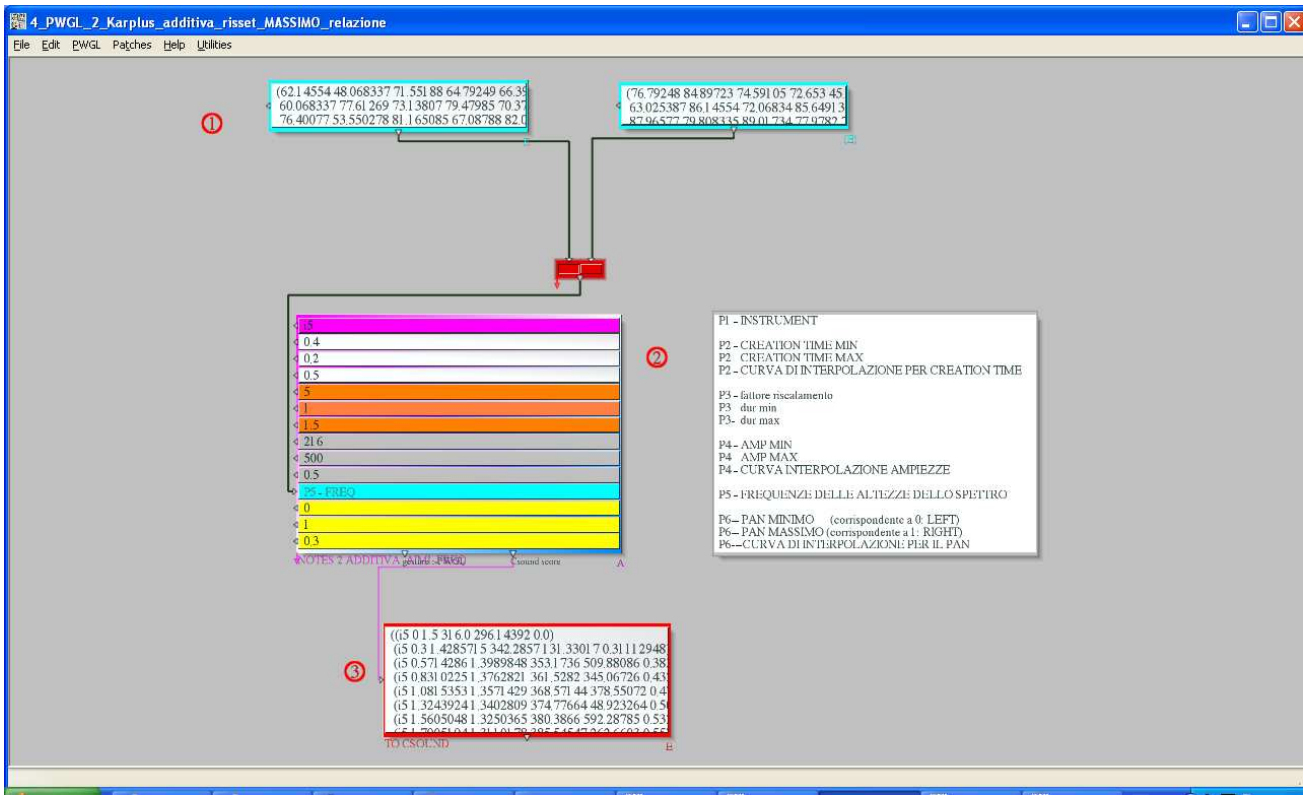


**Figure 2:** The PWGL patch enables transfer of the data in Csound. In 1) the starting (harmonic or inharmonic) spectra can be chosen; in 2) the abstract- box can be seen. The detail is provided by figure 1. In 3) you can see the score related to the spectrum chosen in 1).

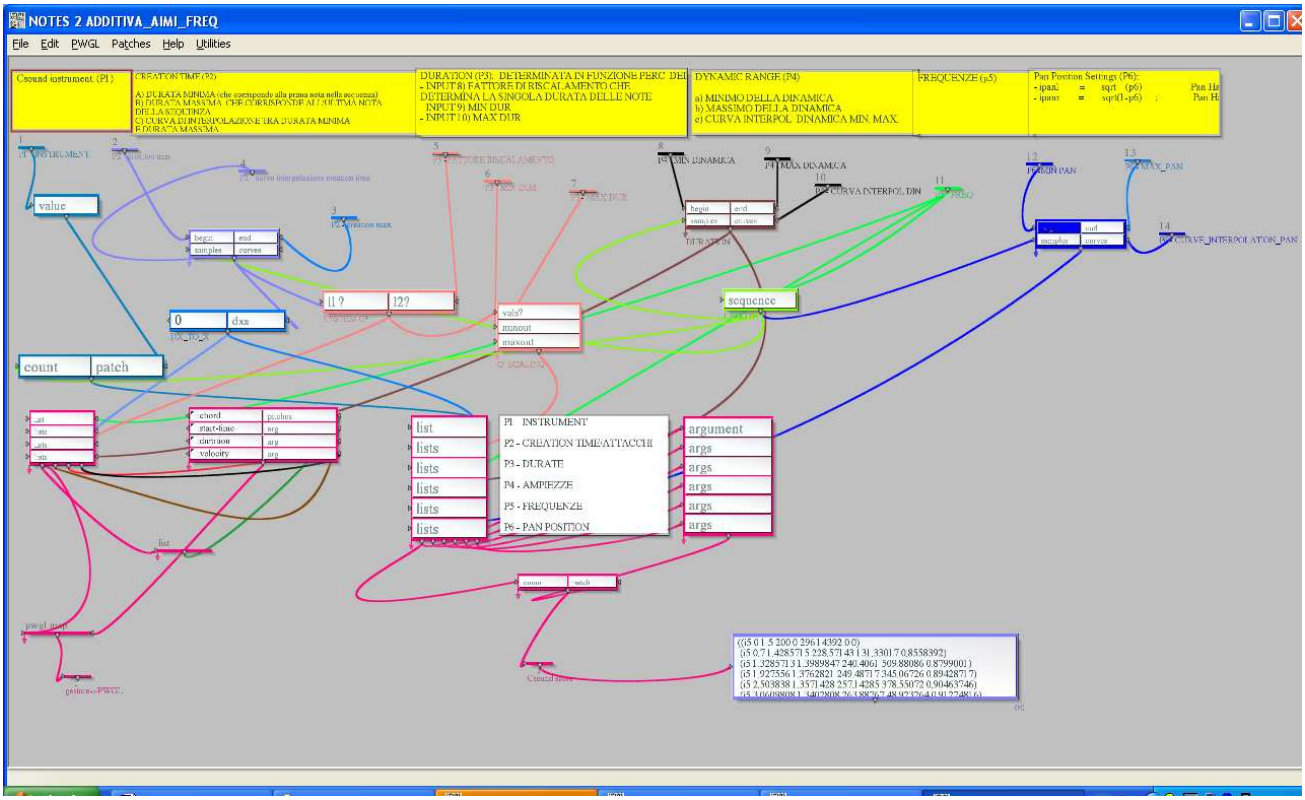


**Figure 3:** The abstract- box is seen from the inside. You can see the entries concerning: 1) instrument; 2) attack; 3) length; 4) amplitude; 5) frequency; 6) pan. In 7) the PWGL-enum box is linked with the first entry of the PWGL-map: if combined they can create loops and they can simultaneously process several links of values, that is to say those concerning parameter-field. The result is placed in a text-box for every interaction. The text-box serves as "result-list": that's the Csound score.

---

Once the sequence of the first notes is set, the patch for the synthesis of Karplus I've just described can be substituted with an additive patch and move from one synthesis technique to another [9], by using, for instance, the following instrument:

```
giamp_fact      = 16     ; Amplitude factor
          Instr 2
idur      =          1/ p3
iamp      =          p4 * giamp_fact
ifreq     =          p5
ipanl  =       sqrt(p6)                ; Pan left
ipanr  =       sqrt(1-p6)              ; Pan right
aamp      oscili     iamp, idur,  2
aout      oscili     aamp, ifreq, 1
outs      aout*ipanl, aout*ipanr
endin
```

---

[8] "The most popular spatial illusions are horizontal panning – lateral sound movement from speaker to speaker – and reverberation – adding a dense and diffuse pattern of echoes to a sound to situate it in a larger space.
Vertical panning (up and down and overhead) can also create striking effects in electronic music [19]."
Mickelson mentions strategies for panning and offers a grading of some basic strategies: Simple Pan; Square Root Pan; Sine Pan; Equal Power Pan; Delay and Filtered Panning [15]

[9] For an in-depth study of the additive synthesis techniques see [11-14] and [16-18].

From these examples it follows that the instruments, after being created, can be employed again in other circumstances, apart from the initial material and the algorithmic- composing material (See figures 9-10).

## 1.1 - ADDITIVE SYNTHESIS

PWGL offers the opportunity to follow the composing process of a historical piece again and to rediscover the conceptual idea by singling out the composing rules and any exceptions.
In the works I've personally rebuilt, my attention has focused on pieces such as Studio II by K. Stockhausen. In this study the German composer uses 81 frequencies starting from the grave limit of 100 hertz.

From this basic frequency he obtains a scale of 80 following frequencies, a scale of 25 identical intervals, from harmonic 1 to harmonic 5, whereas the frequency rate in equable temperament is the twelfth root of two.
Stockhausen employs intervals that are wider than a semitone, with a frequency rate that is the twenty-fifth root of five: each following frequency was obtained by multiplying the previous one by 1.066494.

Five is a recurring number in the piece: 5 notes make a sequence; 5 sequences make a set or group ("sequenzgruppe"). After singling out the frequencies, the lengths in relation to the length of the tape (2.5 centimetres is equal to 0.039 seconds) and the loudness, the different sections of the piece can be rebuilt for each sequence or group: 5 parts plus the final coda [10], as you can see in figure 6 concerning the rebuilding of the first section of the piece.
PWGL enables not only the rebuilding of historical pieces but also the reinterpretation of sounds of the past: this work can be good practice, especially for those approaching Csound for the first time.

In the second example proposed by composer H. Torres Maldonado, we create micropolyphonic textures of spectral nature starting from an instrument designed by J.C. Risset [11] and create, as usual, a patch to transfer the result in Csound.
This lovely sound, similar to the Tibetan harmonic chant, produces an arpeggio with the series of harmonics: you can clearly hear the fundamental and the partial ones from 5 to 9.
Risset's score was modified by using a starting pitch of C2 - 65.41 Hertz instead of the original frequency of 96 Hertz.

The notes which weren't present in the arpeggio will be used to produce new spectra. From the starting frequency of 65.41 Hertz the new set of pitches obtained is (see figure 5): 69.3; 77.78; 87.31; 92.5; 103.83; 110.0;123.47 Hertz.
These pitches, along with the starting frequency, will be the fundamentals of the new spectra of the micropolyphonic texture.

Even in such circumstance the abstract (see figure 8) will collect the music information of the Csound score: instrument; start time, defined by Fibonacci series; length, derived from the retrograde of the series; frequency, amplitude; offset.

[10]It's fundamental to read about the score [22] and the analysis of H. Silberhorn [20].
[11]The instrument, created by the French composer Jean Claude Risset for the piece "Mutation", is included in The Amsterdam Catalog of Csound Computer Instruments, available on the website http://www.music.buffalo.edu/hiller/accci/02/02_43_1.txt.html [16]

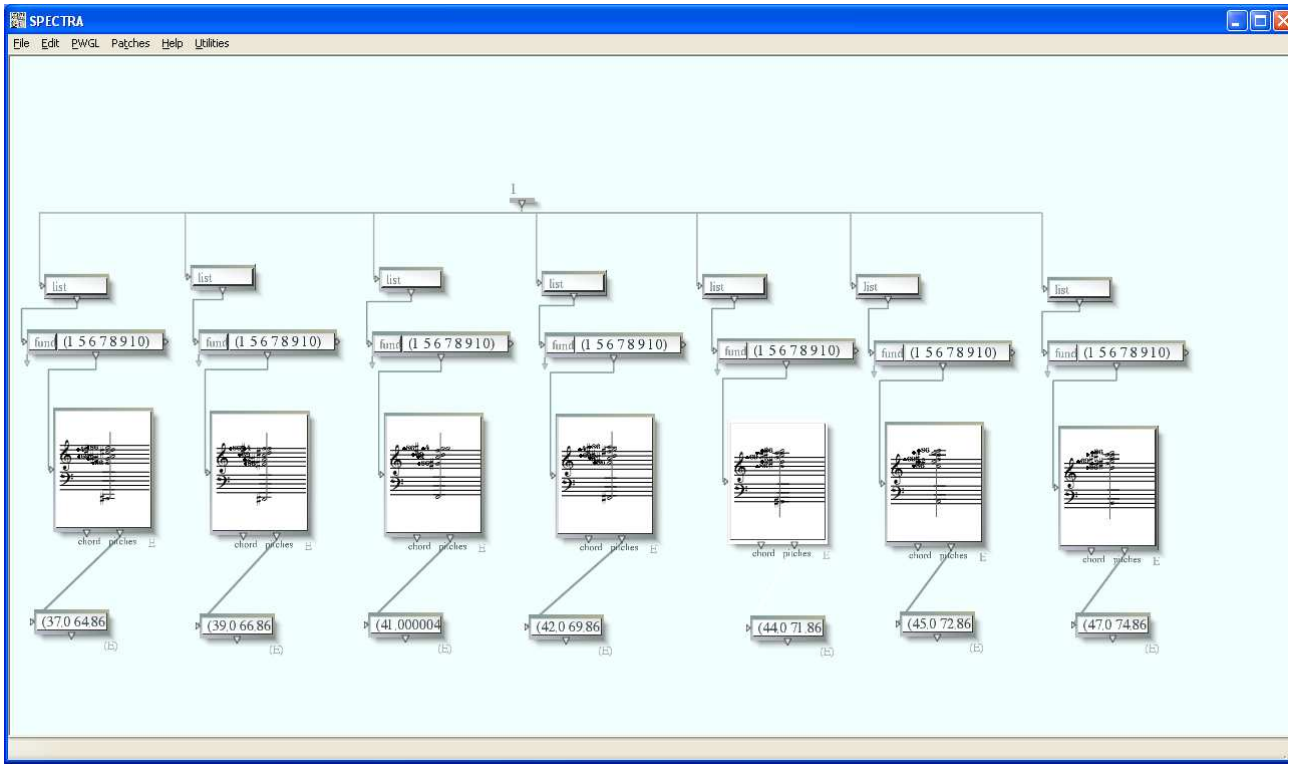


**Figure 4:** production of new spectra with fundamental 69.3;77.78; 87.31; 92.5; 103.83; 110.0; 123.47

```
F1 0 1024 10 1 0 0 0 .7 .7 .7 .7 .7 .7
;      start   dur    freq    amp     offset
i1     1       68     65.41   1500    .03
i1     2       42     69.3    1500    .03
i1     3       26     77.78   1500    .03
i1     5       16     87.31   1500    .03
i1     8       10     92.5    1500    .03
i1     13      6      103.83  1500    .03
i1     21      4      110     1500    .03
i1     34      2      123.47  1500    .03
endin
```

**Figure 5:** Music information of the Csound score.

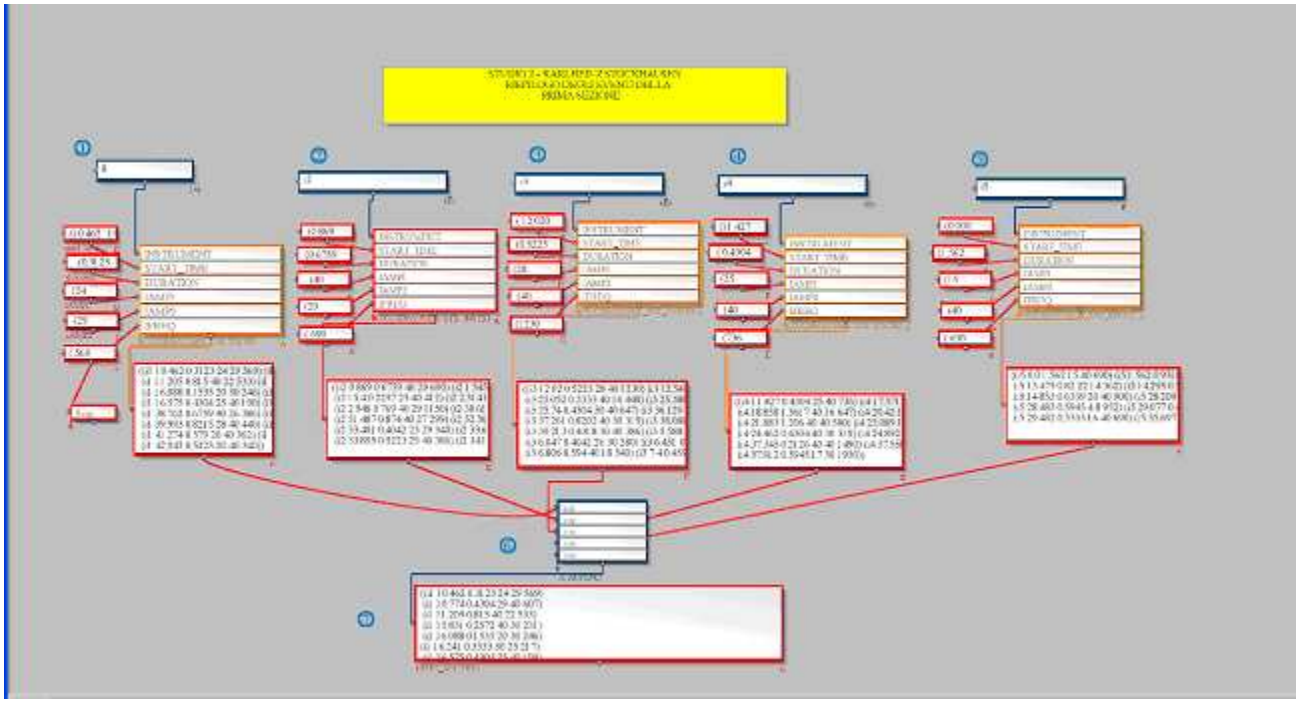

**Figure 6:** First section of Studio II. In (1-5) you can see the material for the five instruments or "mixture" composing the "gruppen". In the result list (7) you can see the final score achieved by gathering the x-append of the score for each instrument (6).

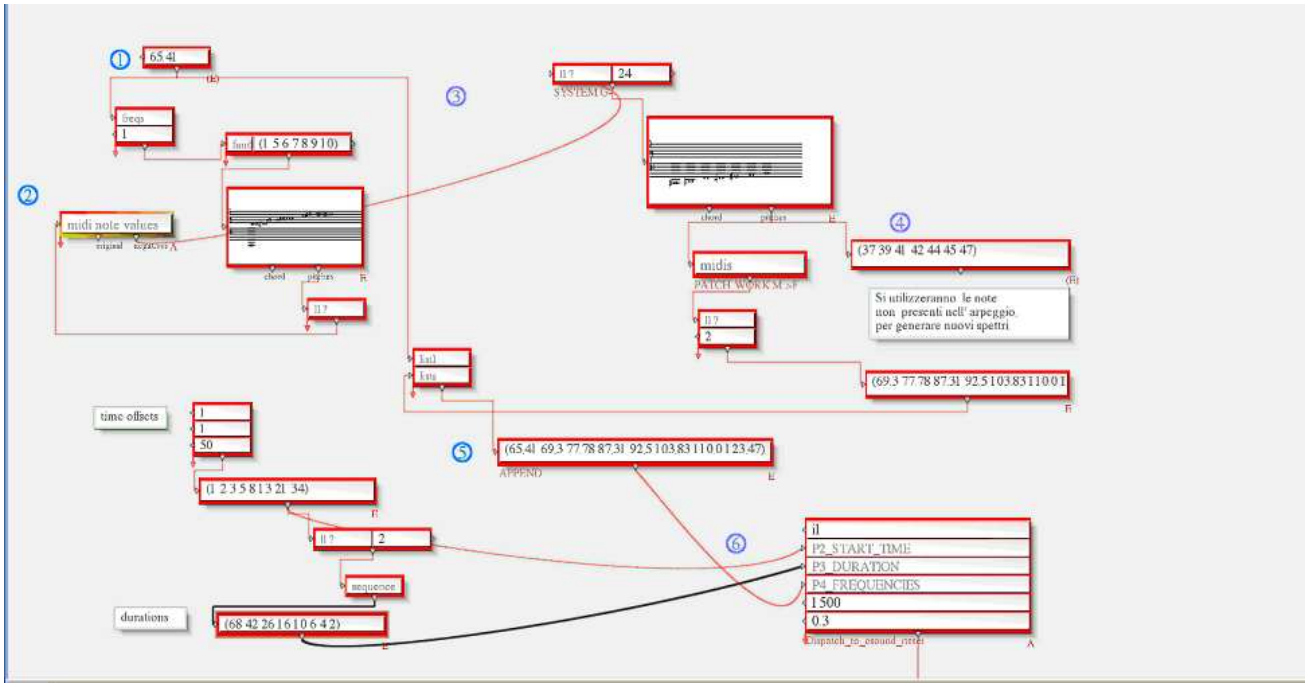


**Figure 7:** in (1) you can see the fundamental frequency of the arpeggio (65,41 Hertz) and relative harmonics (2). In (3) the spectrum is made "negative" by obtaining a set of frequencies that will pave the way for new arpeggios (4). In (5) the fundamental frequency is added to the resulting frequencies, which brings about the abstract- box (6).

[12] I agree with M. Stroppa: "Since the aesthetic needs of a musician cannot be guessed every attempt at searching for a more general solution must generate a System which is both as open as possible and very easy to personalize. The musician's first task will then be to adapt it to his or her own particular way of thinking about sonic potential"[23].

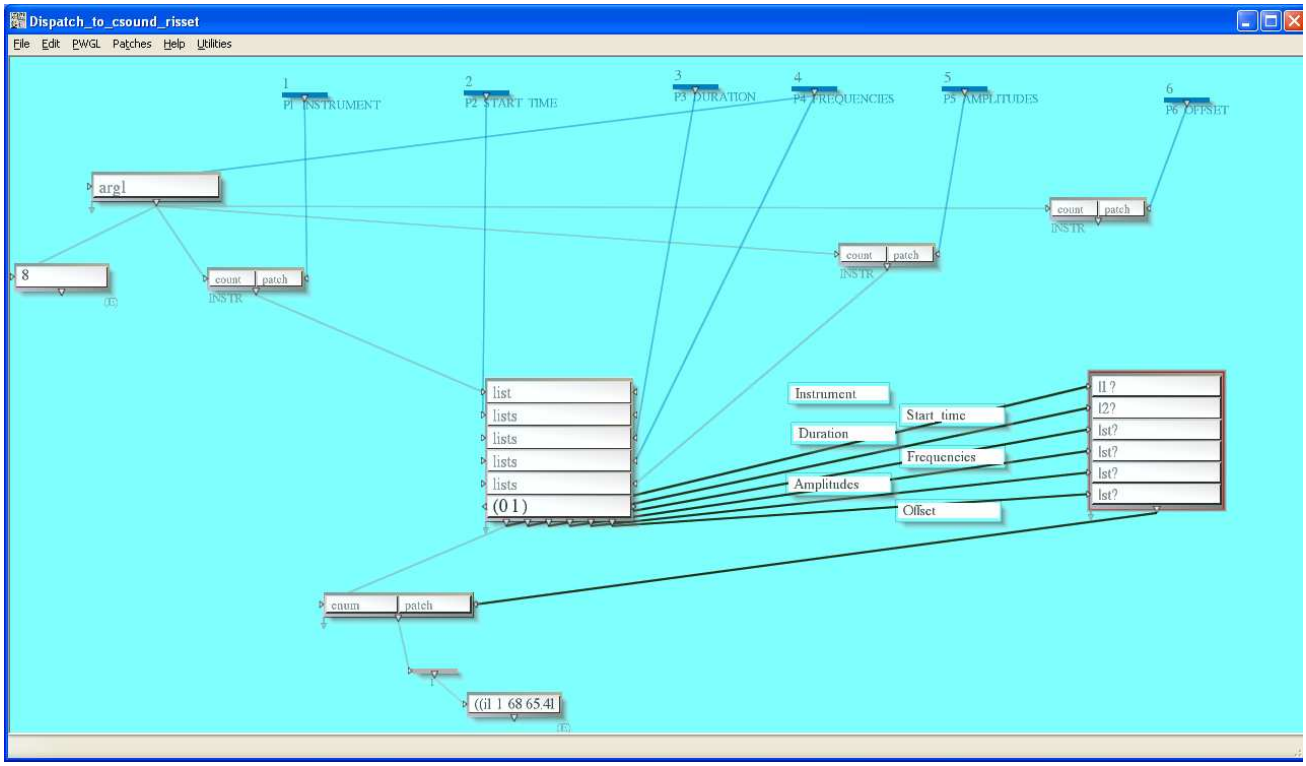


**Figure 8:** Detail of the abstract- box in figure 6: all the music data to be tranferred to Csound can be seen.

.

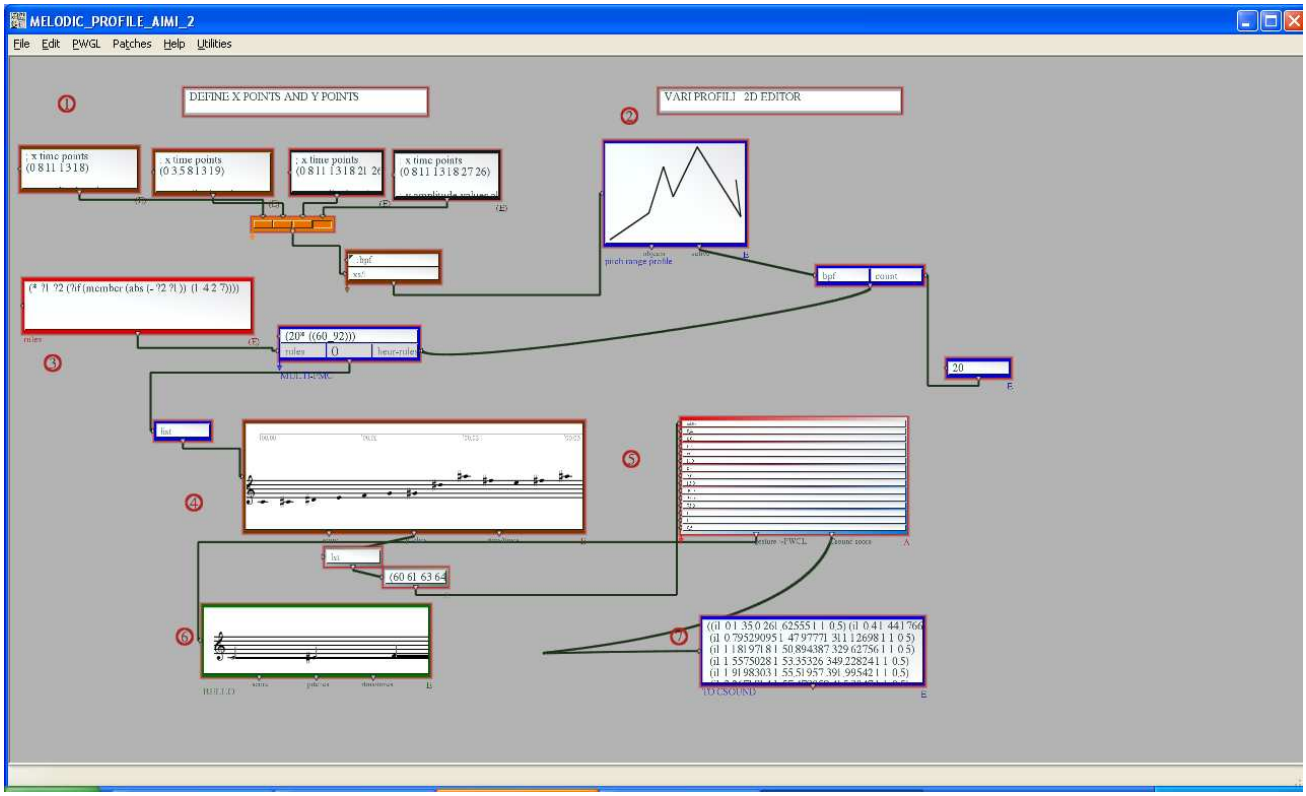


**Figure 9:** A sequence of midi notes between the extremes follows a melodic profile chosen among the ones available (2). The melodic profile is made through Cartesian values (x,y) in a text-box (1). The result is influenced by the type of interval chosen by using the heuristic rules defined in (3). The resulting sequence is accelerated or decelerated and provides a sequence in (6) with the relevant score in Csound (7).

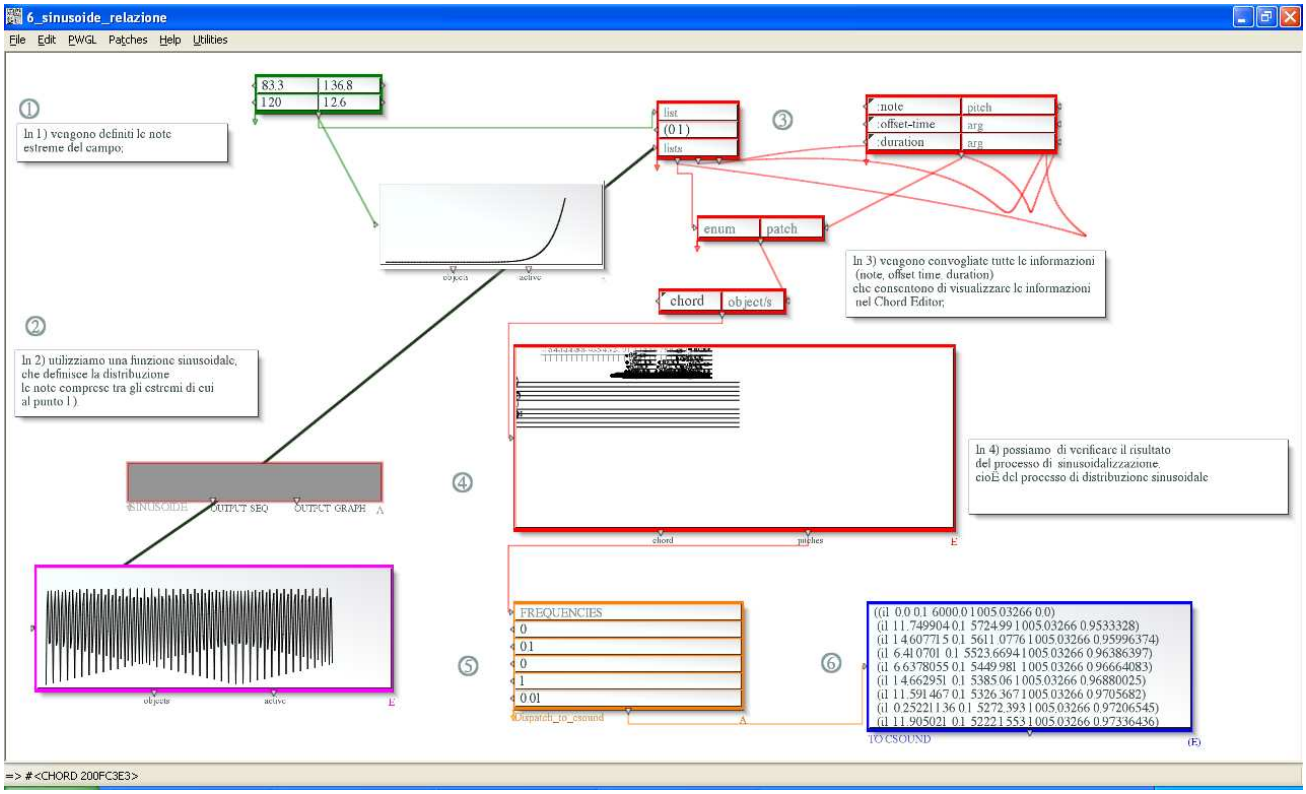


**Figure 10:** A sequence of notes defined between two extremes follows a sinusoidal harmonic profile combined with a concave/convex curve. In 1) the extreme notes of the sequence are defined; in 2) a sinusoidal function is used to deliver the notes; in 3) we interpolate, by using PATCH-WORK INTERPOLATION, between the two. In 3) all the information (note, offset time, duration etc.) is channeled, making it possible to see the information in the "Chord Editor"; in 4) we can check the result of the sinusoidal delivery process. In 6) we get the result to be transferred to Csound by means of an abstract-box (5).

---

# CONCLUSION

In this article I've described how to blend PWGL and Csound, in order to make a collection of Csound instruments. By doing so, musicians can avoid using expensive programs and libraries, which are often influenced by the ideas and approaches - even the composing ones – of their creators, and they can make their own.

Thanks to these modest suggestions, musicians can create their own libraries for sound synthesis and can modify them[13], on the basis of their needs as these present themselves.

I've described the first result of a work in progress which will lead to a suite of synthesizers or a virtual synthesizer to conduct orchestras and Csound synthesis; such instruments will come up alongside the existing PWGLSynth modules. PWGL will become an environment for the design and graphical display of the instruments.

It's easy to envisage that PWGL will have an increasingly important role in algorithmic composition and sound synthesis, thanks to the support of an ever wider community of developers and final users.

[13] I agree with M. Stroppa: "Since the aesthetic needs of a musician cannot be guessed every attempt at searching for a more general solution must generate a System which is both as open as possible and very easy to personalize. The musician's first task will then be to adapt it to his or her own particular way of thinking about sonic potential"[23].

# 4. Acknowledgements/references


[1] Association of Lisp Users (A.L.U), http://www.lisp.org/alu/home.

[2] R. Boulanger, *The Csound Book - Perspectives in software synthesis, sound design, signal processing, and programming*, R. Boulanger Ed., p. 370, 2000.

[3] R. Bianchini , A. Cipriani, *Il suono virtuale - Sintesi ed elaborazione del Suono – Teoria e pratica con Csound*, pp. 343 - 362, 2007.

[4] M.Goggins, *A Csound Tutorial*, pp. 54-55, 2006.

[5] D. Jaffe, J.O. Smith "Extensions of the Karplus-Strong Plucked-String Algorithm", *Computer Music Journal 7(2),* 1983. Reprinted in C.Roads The

[6] Music Machine. MIT Press, pp. 481-494, 1989.

[7] K. Karplus, and A. Strong "Digital Synthesis of plucked string and drum timbres", *Computer Music Journal 7 (2)*: pp. 43-55, 1983.

[8] M.Kuuskankare, M. Laurson. "PWGL, Towards an Open and Intelligent Learning Environment for Higher Music Education", *Proceedings of the 5th*

*European Conference on Technology Enhanced Learning,* EC.- TEL 2010 – Barcelona, Spain, p.520, 2010.

[9] M. Kuuskankare, M. Laurson, V. Norilo. "PWGLSynth, A Visual Synthesis Language for Virtual Instrument Design and Control", *Computer Music Journal*, vol. 29, no. 3, pp. 29-41, 2005.

[10] M. Laurson and V. Norilo. "Copy-synth-patch: A Tool for Visual Instrument Design", *Proceedings of ICMC04*, Miami, 2004.

[11] M. Laurson and V. Norilo. "Recent Developments in PWGLSynth", *Proceedings of DAFx 2003*, pp. 69-72, London, England, 2003.

[12] D. Lorrain,"Inharmonique, Analyse de la Bande de l'Oeuvre de Jean-Claude Risset", *Rapports IRCAM*, 26, 1980.

[13] M. Mathews, J.-C. Risset. *"Analysis of Instrument Tones"*, Physics Today 22(2): pp. 23-30, 1969.

[14] F.R. Moore, 1977, 1985. "Table Lookup Noise for Sinusoidal Digital Oscillators", *Computer Music Journal 1(2)*:26-29.

Reprinted in C. Roads and J. Strawn, Foundations of Computer Music. MIT Press, pp. 326-334, 1985.

[15] J.A. Moorer, *"Analysis-based Additive Synthesis"* in Strawn, J. Digital AudioSignal Processing: An Anthology. A-R Editions, pp. 160-177, 1985.

[16] H. Mickelson "Panorama" , *Csound Magazine,* Autumn 1999.

[17] J.-C. Risset, *An Introductory Catalogue of Computer Synthesized Sounds*, reprinted in "The Historical CD of Digital Sound Synthesis", Computer Music Currents nº 13, Wergo, Germany, 1969.

[18] J.-C. Risset, *Additive Synthesis of Inharmonic Tones* in M.V. Mathews and J.R. Pierce, eds. 1989. Current Directions in Computer Music Research. MIT Press, pp. 159-163, 1989.

[19] J.-C. Risset, *Computer Music Experiments 1964-...* in C. Roads, “The Music Machine"” MIT Press, pp. 67-74, 1989.

[20] C. Roads, *Computer Music Tutorial*, MIT Press, p. 452, 1996.

[21] H. Silberhorn, *Die Reihentechnik in Stockhausens Studie II*, 1980.

[22] G.L. Steele, *“Common Lisp The Language”*, 2nd Edition. Digital Press, 1990.

[23] K. Stockhausen, *Nr.3 Elektronische Studien, Studie II, Partitur*, Universal Edition, Zürich-London, 1954. Wien

[24] M. Stroppa, *Paradigms for the high-level musical control of digital signal Processing*, Hochschule fur Musik und darstellende Kunst Stuttgart, Germany, p.2, 2000.