Towards a Borgean Musical Space: An Experimental Interface for Exploring Musical Models

Charles de Paiva Santana, Jônatas Manzolli, Jean Bresson, Moreno Andreatta

To cite this version:

Charles de Paiva Santana, Jônatas Manzolli, Jean Bresson, Moreno Andreatta. Towards a Borgean Musical Space: An Experimental Interface for Exploring Musical Models. Electronic Visualisation and the Arts, 2015, London, United Kingdom. 10.14236/ewic/eva2015.8 . hal-01177456

HAL Id: hal-01177456
https://hal.science/hal-01177456

Submitted on 17 Jul 2015

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

http://dx.doi.org/10.14236/ewic/eva2015.8

Towards a Borgean Musical Space: An Experimental Interface for Exploring Musical Models

Charles de Paiva Santana[1,2]
Charles.de.Paiva@ircam.fr

Jônatas Manzolli[1]
jonatas@nics.unicamp.br

Jean Bresson[2]
Jean.Bresson@ircam.fr

Moreno Andreatta[2]
Moreno.Andreatta@ircam.fr

[1)]UNICAMP
Núcleo Intedisciplinar de Comunicação Sonora
Rua da Reitoria 165, Campinas, Brasil

[2)] UMR STMS
IRCAM – CNRS – UPMC
1, place Igor Stravinsky, 75004 Paris, France

Musical compositions can be the object of advanced computational modelling. We develop this idea by creating formal models allowing the study of musical works from the repertoire and conceiving corresponding computational tools. From this perspective, one can aim to model, produce and evaluate different instances and configurations of such models. In this paper we discuss an experimental tool, as an attempt to visualise the space of instances that can be generated from a musical piece's model. This tool consists in an interface attached to a class in the OpenMusic environment, where parameter values are compared with the result of analytical descriptors, as that of 'cognitive-sonance' from the "Sonic Object Analysis Library" (SOAL). We demonstrate how this interface works by exploring it with a previously built model of James Tenney's piece, the Spectral Canon for Conlon Nancarrow.

Music analysis. Modelling and simulation. Music composition. OpenMusic. James Tenney. Spectral music.

1. INTRODUCTION

It was demonstrated that existing musical scores can be rebuilt or generated using computer models (Riotte & Mesnage 2006) and it was suggested that neighbouring variants can also be produced by such procedure:

> [...] for us, to model a musical score is to model the composition process by an algorithm able to reproduce, either the score or the neighbouring variants obtained by a different set of parameters. (Rokita 1996).

However, it was not clear then if these variants were of any musicological or creative interest.

Our premise is that a musical score is one single occurrence of a system's particular configuration. Musical variants are envisaged by modelling the behaviour of such systems and by modifying its parameters values.

We call those variants the different instances of a piece. Being similar or unlike, instances are ontologically related. Their study results in further knowledge about the musical work's inherent attributes and open for new creative possibilities. Hypotheses about a musical composition's structural attributes can be verified by simulation and conclusions can be made methodically. The original score could prove to be, or not, an optimal occurrence of the system.

The model is a computer program consisting of a group of subroutines that allows the reconstruction of the musical score or some of its specific aspects. We test different settings of such model, generating and analysing instances for explicative and speculative means. The model inputs are parameter values, its output is symbolic musical data. The subroutines implement music composition techniques and related musical tasks.

Aspects of a musical score that reflect compositional "choices" are implemented as parameters. The model conception is based in preliminary musical analysis of the chosen piece. The analysis of the musical score and its different instances generally follows musicological practice.

In this paper, we discuss a tool we developed as a first attempt to address the large number of variants a model can produce. Argentinian writer Jorge Luis Borges in his short history “The Library of Babel” envisaged a space that is finite but contains a countless number of instances. We try with our tool to visualise such a space and create relationships between the instances figured in it.

For this purpose, we use a model we built previously, reconstructing James Tenney’s piece *Spectral CANON for CONLON* Nancarrow (see De Paiva Santana et al. 2013).

2. THEORETICAL FRAMEWORK

Musical and extra-musical models, as well as the use of modelling, played different roles through the evolution of most civilisations’ music practice, displaying rich interactions between human knowledge and human activity. André Riotte and Marcel Mesnage remark that the concept of model "cover a entire hierarchy of attitudes and operations", the concept being always underlying in music history (Riotte and Mesnage 2006). Examples as old as the Pythagorean and Chinese scales (mathematics), the construction of the organ and violin (acoustics) or the musical initiation through the àkonés in Central Africa and the Indian bandish (pedagogy and cognition) exist among many others. In western music composition, we identify the use of models from the medieval canon and isorhythm to the utilisation of mathematical models in the music of Iannis Xenakis and Jean-Claude Risset. Acoustic models are used in the so-called spectral music.

According to the tripartite system proposed by Jean Molino, in the case of musical composition "it makes sense to speak, for a work or part of a work, of musical system and to focus on the complexity of that system" (Levy 2004). Modelling, a recurring theme in computer music, aims to "formalize various aspects of musical phenomena and to construct the corresponding computer models, with the objective of understanding or analysis, at first, then often creating and producing music" (Bresson 2007).

2.1 Concept of Modelling and Simulation in the present work

In our work, we consider a model as a “simplified representation of a complex process which allows a simulation conforming the reality within given limits” (Rokita 1996).

The reconstruction of the musical score is a simulation of the musical composition that is not necessarily faithful to the real act of the composer (or the poietic level – see next section) (ibid.). Such simulation can serve the purposes of either verification and observation (explicative) or the production of similar processes (creative) (Assayag 1993).

2.2 Modelling and Tripartition

From a semiological position, our modelling is based mostly on the immanent properties of the musical score (its neutral level), rather than how the composer made the score (poietic level) or how the piece is perceived by the listener (esthesic level).

The modelling of musical scores can though relate to all levels, as they may not be independent from each other. From Molino's tripartition, Jean-Jacques Nattiez devised six different categories of musical analysis (Figure 1) from which correspondent categories of score modelling can be envisaged (see Nattiez 1987).

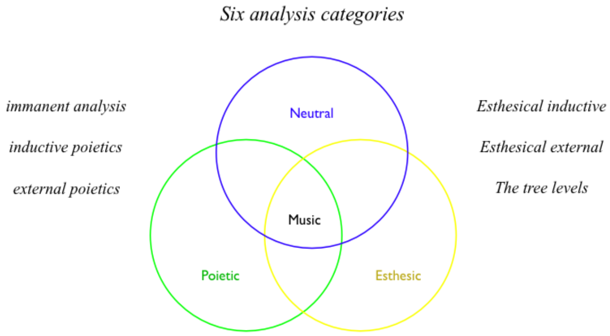


Figure 1: *Nattiez’s six categories of analysis based on Molino’s Tripartion: the neutral level, the poietic level (composer’s perspective) and the esthesic level (reception)*

2.3 Methodology

Our work on the modelling of musical pieces can be separated into three main steps: 1) the formalisation (which includes musical analysis in a broader sense), 2) the elaboration of a corresponding computer model and 3) the exploration of the model (the simulation of compositional decisions and experimentation of variants).

Some pieces were chosen in the very beginning as 'simple' case studies. We opted for pieces displaying a higher degree of clarity in their formal processes, as seen through the reading of the scores or in their audition. In this paper we shall work with an algorithmic piece composed by James Tenney.

We use OpenMusic computer-aided composition environment (Bresson et al. 2011), which offers a large set of ready-to-use *functions,* performing musical tasks and where new routines can be easily programmed and integrated. In this environment, the output of the model can be heard as an audio or MIDI file, seen as a musical score and fine-tune edited.

3. SIMULATION OF COMPOSITIONAL DECISIONS IN A PIECE BY JAMES TENNEY

The Spectral CANON For CONLON Nancarrow is a piece for player piano by American composer James Tenney (1934 – 2006) based on the idea of a correspondence between duration and frequency interval ratios. The piece consists of a 24 part canon, where all voices share a same series of decreasing durations (accelerando), which is then reversed (forming a palindrome), and are superimposed following a precisely determined pattern. In Tenney's original version, the piece suddenly stops when a point of synchronism is reached. Another version, conceived in collaboration with composer Clarence Barlow, uses the same specifications as the original but the piece stops when each voice forms a complete palindrome. Our examples in this paper follows Barlow's version.

The key to understand the canonic construction and its uniqueness rely in one idea presented in the influential book *New Musical Resources* (1930) by Henry Cowell. It consists in a way to correspond durations and frequencies where each element in a *series of partials* multiplies the frequency and the number of rhythmic beats of a given fundamental. In the example of Figure 2, the second partial of a fundamental 16 Hz plays twice its frequency and its beats, the third partial thrice and so forth. It can also be viewed as an illustration of the mechanism behind the *Rhythmicon*, a keyboard instrument capable of playing rhythmic ratios with corresponding pitches invented by Cowell.

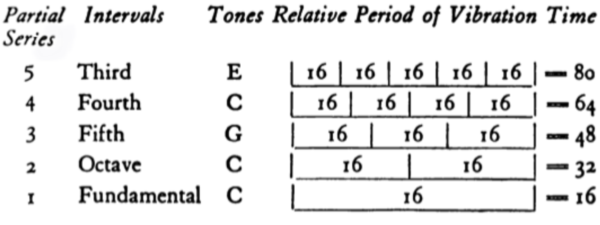


Figure 2: *Henry Cowell's model to match frequencies and durations based on the harmonic series. Reproduced from New Musical Resources (1930, p.47). It proposes a model to correspond frequency and durations based on the harmonic series.*

Cowell called 'rhythmic harmony' the selection and superposition of durational ratios in an analogy to the construction of chords based in the harmonic series. The C major chord can be formed from the third, fourth and fifth overtones represented as 5:4:3. He proposed that the same proportion could be rendered as a rhythm. James Tenney used a similar idea when constructing his piece as shows Figure 3. The mathematical formalism that constitutes the base of our model is shown in Figure 4.

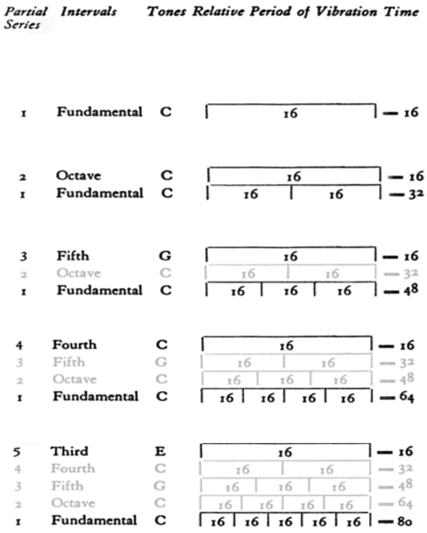


Figure 3: *Cowell's original figure modified to show how Tenney adapted the Rhythmicon idea. The fundamental multiplies the rhythmic beats of each new partial*

Starting from the implementation of the canon's generative process, we built a global model highlighting a number of parameters: first duration of the series, frequency fundamental, number of voices and initial ratio. This model corresponds to the simplest and most explicit generative process, reduced to its essential number of parameters.

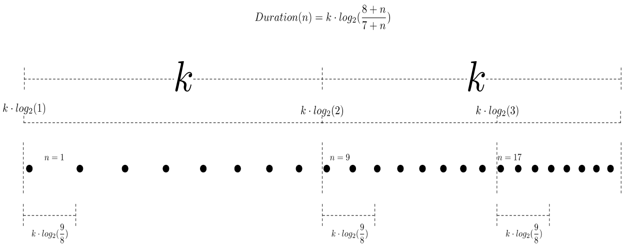

$Duration(n) = k \cdot log_2(\frac{8+n}{7+n})$
k
k
$k \cdot log_2(1)$
$k \cdot log_2(2)$
$k \cdot log_2(3)$
$n = 1$
$n = 9$
$n = 17$
$k \cdot log_2(\frac{9}{8})$
$k \cdot log_2(\frac{9}{8})$
$k \cdot log_2(\frac{9}{8})$

Figure 4: *Reduced formal representation of the series of duration shared with each canonic voice in Tenney's piece. K is a constant representing the value of a "durational octave" from which the first duration of the series (and the first duration of the piece) can be controlled (Source: De Paiva Santana et al. 2013)*

Different values for each of the model's parameters were tested. Although we constructed our model from a "neutral perspective", considering mostly its immanent properties, we were able to make assumptions about the composer creative choices, which would correspond to a poesis inductive analysis, from the neutral level to the poietic.

By varying the first duration of the piece, we have concluded that around four seconds, the value used by Tenney is ideal for the listener not to loose interest in the piece, avoiding extremely long durations in the beginning and to not exceed the threshold of onset differentiation in its densest part.

One of the parameters which has the strongest effect in the musical texture is the initial ratio of the series of durations. An instance of the model generated using an initial ratio of 2:1 (Figure 5) rather than 9:8 for the durations leads to an equally well-structured piece but its texture in the “forward segment” is more “choral” than the polyrhythmic one Tenney intended in his homage to Nancarrow. Likewise, keeping the initial 9:8 ratio for all the series (Figure 6), the pitch ambitus becomes too narrow and the canon looses most of its timbre richness and perceptual features.

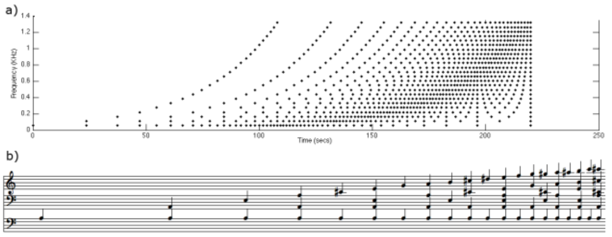


***Figure 5:** An instance of the canon with initial ratio of 2:1 generated from our model. In this instance the musical texture does not feature the polyrhythms of the original version (Source: De Paiva Santana et al. 2013)*

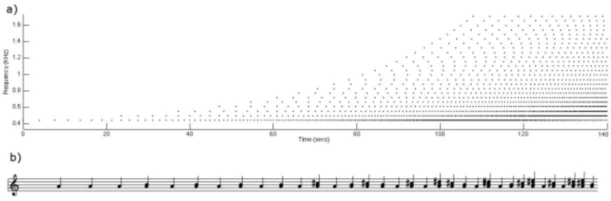


***Figure 6:** An instance of the canon where the pitches and the series of durations begin with the same ratio (9/8); the piece’s texture becomes very narrow (Source: De Paiva Santana et al. 2013)*

Subsequently, we conceived an extended version with additional parameters and features not designed by James Tenney. This model would correspond to a more experimental approach. Figures 7 and 8 show instances of the canon, with the inclusion of a ‘distortion’ process, which proportionally stretches the durations and frequencies of the canon. The details of our exploration and discoveries of these experiments are illustrated and commented in the aforementioned publication (de Paiva Santana et al. 2013).

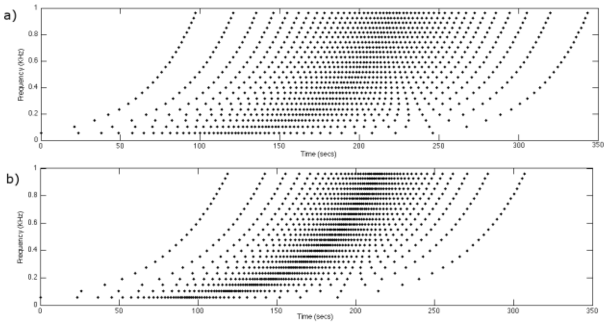


***Figure 7:** Slight distortions of the canon. a) dist = 0.9; b) dist = 1.1 (Source: De Paiva Santana et al. 2013)*

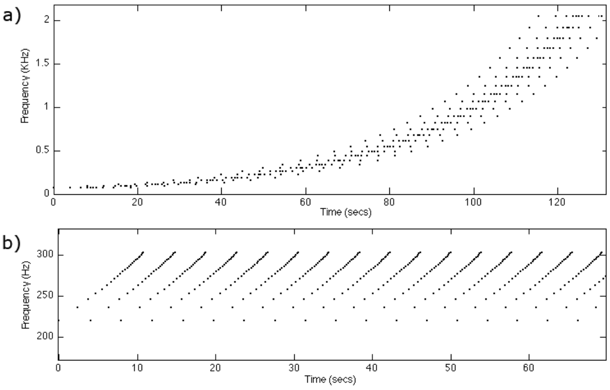


***Figure 8**: Extreme distortions of the canon. a) dist = 2.5; b) dist = 0.1 (Source; De Paiva Santana et al. 2013)*

4. EXPLORATION OF THE SPACE OF THE MODEL INSTANCES

In our previous examples we used a 'diacronic' representation of the model instances; how its pitch and durational content evolves in time. With this relatively traditional way of representing the instances we were able to appreciate the singularities of a single or small set of parameters variation. We would like, as a further step in our exploration, to show each instance as a point in a space and make relations between a greater number of them.

Such approach has the advantage of demonstrating the 'sensibility' of each parameter, how each of them has an effect in specific characteristics of a large group of instances.

Our idea was to consider those instances ‘outside time’ and then visualise this space with a graphical interface. We call this tool the *Model Xplorer*. In its graphical interface one can visualise 2D or 3D plots relating analytical measurements and parameters values on top of the possibility of listening and visualising the score of each instance. It was conceived to be adaptable to other models.

We have tested this tool with our implementation of Tenney's piece discussed above, which is perhaps the simplest model we built in terms of computational power required to generate a large number of instances (see Shvets and De Paiva Santana 2014, for a collaborative modelling of Arvo Pärt's *Spiegel im Spiegel*).

The *Model Xplorer* is a special class in the OpenMusic visual programming environment, which takes as an input the functional implementation of the model, and as another input a list of values representing the model parameters and analytical data. The analytical data is the output of descriptors computed on generated instances that can give to the user an estimation of some characteristics these instance.

The tool works in two steps. In a first step a function called "generate-instance" requires a set of values for each parameter to be passed to a subroutine containing the model implementation and the analysis modules. The creation of MIDI files for each instance is possible inside this subroutine.

The function "generate-instance" then saves each instance's parameter and analytical values in a text file in the computer hard disk.

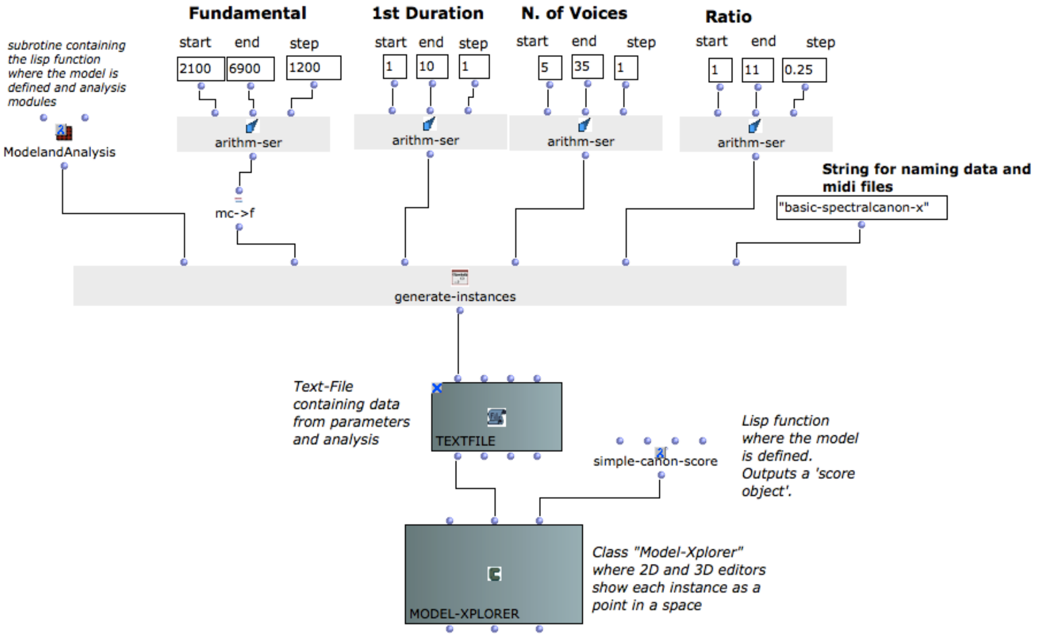
For the examples in this paper, we considered the four parameters of our first implementation of Tenney's piece: fundamental, first duration, number of voices and initial ratio. For the analytical modules we considered the value of the 'last onset', 'mean-frequency and 'cognitive-sonance'.

The first two analytical measurements were chosen as a very simple way to evaluate how the pitch and durational content of the instances change with parameter variation. For instance, as our model is deterministic (no stochastic process is involved), one would expect that the more voices the canon has, the higher is the mean-frequency and the bigger is the last onset, showing a linear relationship. That being the case, the *Model Xplorer* functions as a way of validating the consistence of the model construction, each parameter variation reflecting a 'reasonable' equivalent effect on the piece's features.

On the other hand, the measurements of the 'cognitive-sonance' module should show a less linear relationship with parameter variation. This analytical function is part of a library developed for OpenMusic called Sonic Object Analysis Library (SOAL). Its description is given as the following:

> A vector from "maximum consonance" to "maximum dissonance", evaluates the contiguous dyads of the input file (s) on the basis of a weighting of their relative dissonance, according to a pre-defined cognitive/cultural background, and extracts an average value for the whole MIDI file, its "sonance" rate. (Guigue 2010).

In other words, "sonance" indicates how consonant or dissonant a musical file could be perceived according to a weighted table of intervals given as an input based on "cognitive or cultural



subrotine containing the lisp function where the model is defined and analysis modules

Text-File containing data from parameters and analysis

Lisp function where the model is defined. Outputs a 'score object'.

Class "Model-Xplorer" where 2D and 3D editors show each instance as a point in a space

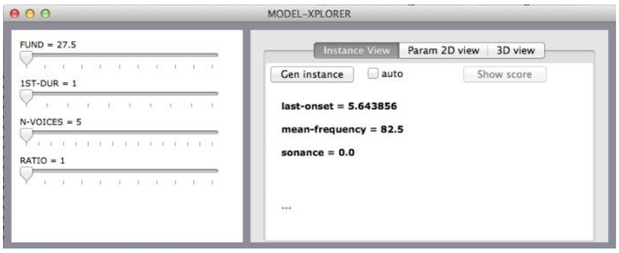
***Figure 9**: OpenMusic patch with all the functions and generation of parameter values required by the 'Model Xplorer ' class; the list of parameter values is controlled by the function 'arithm-ser'; with the values showed in the patch, about sixty thousand instances are generated*

background". For our experiments we opted to use the function's default table (see Guigue 2010).

Figure 9 shows the OpenMusic visual program with all functions required by the class 'Model Xplorer'. The parameter values are generated by the function 'arithm-ser'. It outputs a series based on an *initial value*, an *end value* and a *step*. In this example only five values are calculated for the fundamental (2100 to 6900 by 1200), while there are ten different values to be used as first duration (1 to 10 by 1), 31 for number of voices and 41 for initial ratios. With these settings about sixty thousand instances are generated in one evaluation.

'Generate-instances' passes every parameter value in a loop to the subpatch called 'ModelandAnalysis', which contains the functional implementation of the model and the analytical modules. This process could take hours or days to calculate; some information about the current state and results is printed in the OpenMusic console at each step.
The resultant text file is then passed as an input to the 'Model Xplorer' class, which also requires the functional model implementation in order to generate instances inside the interface.

Figure 10 shows the default interface of the *Model Xplorer*. On the left pane the parameters are represented in movable sliders. Clicking the button "gen instances" will generate a new instance and show a compact representation in the bottom of the right pane. It can also show the score in the appropriate OpenMusic 'score object' editor. Figure 11 shows a window after an instance is generated with some parameter values selected inside the *Model Xplorer* interface.



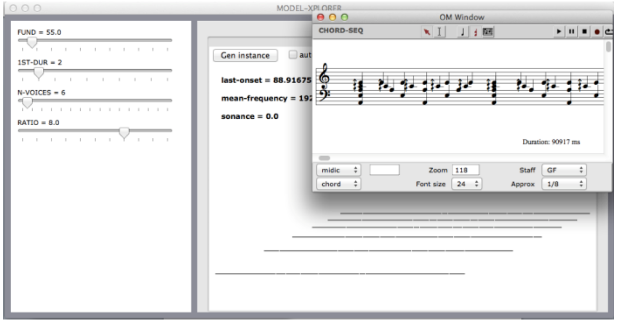
FUND = 27.5
1ST-DUR = 1
N-VOICES = 5
RATIO = 1

last-onset = 5.643856
mean-frequency = 82.5
sonance = 0.0

Figure 10: *Model Xplorer default view. In the left pane parameters are selected using sliders*

Figure 12 now shows a 2D pane illustrating how the parameter 'number of voices' is related with the 'cognitive-sonance' of each instance in a somewhat unpredictable way, while Figure 13 shows how the number of voices has a logarithmic relationship with the values of the last onset.

Note that while the number of voices is fixed (the slider is disabled), the other parameters can still be modified in the left-hand pane.



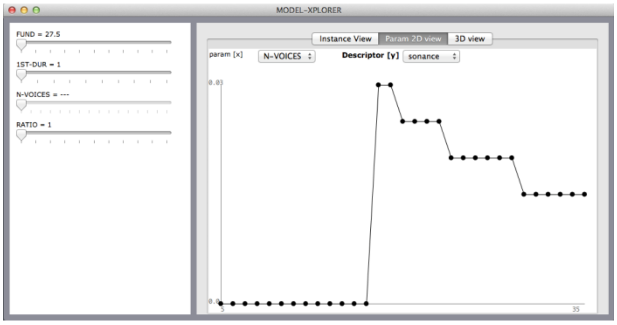
FUND = 55.0
1ST-DUR = 2
N-VOICES = 6
RATIO = 8.0

last-onset = 88.91675
mean-frequency = 19
sonance = 0.0

Duration: 90917 ms

midic
chord
Zoom 118
Font size 24
Staff GF
Approx 1/8

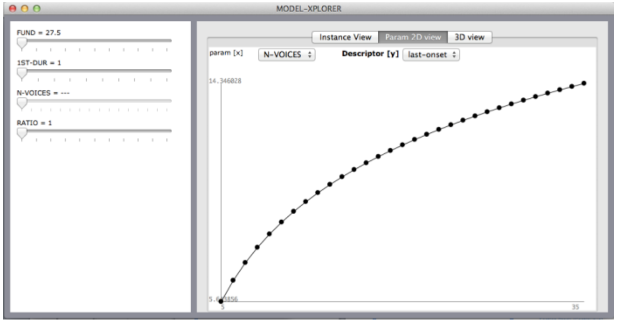
Figure 11: *View of Model Xplorer window after an instance is (re) generated and showed in a score object (smaller window). A preview of the instance is shown in the bottom of the main window*



FUND = 27.5
1ST-DUR = 1
N-VOICES = ---
RATIO = 1

param [x] N-VOICES
Descriptor [y] sonance

Figure 12: *2D pane showing how the number of voices affects the 'sonance' rate of a set of instances*



FUND = 27.5
1ST-DUR = 1
N-VOICES = ---
RATIO = 1

param [x] N-VOICES
Descriptor [y] last-onset

Figure 13: *2D pane showing how the number of voices relates with the value of the 'last onset' in a group of instances*

Figures 14 and 15 finally show the 3D pane of the *Model Xplorer* interface. Two parameters are chosen against one measurement and represented in the horizontal and vertical axes. The descriptor's values are visualised as shades of grey (the darker a cell, the higher the value of the descriptor). We can observe that both the fundamental values and the number of voices affects the mean-frequency, in the same way as both the first duration and the number of voices affect the value of the last onset.

As an experiment, we modified our model to change the value of the '*k*' constant (cf. Figure 4) each time an instance is created. Using a similar principle, we also incorporated a function to 'stretch' proportionally the intervals between the frequencies. Those modifications were made using random numbers and were programed to affect 'moderately' the features of the instances.

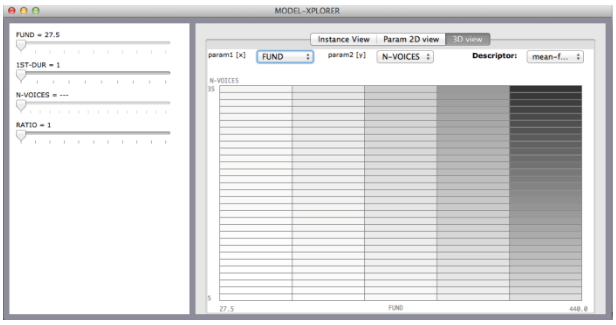

FUND = 27.5
1ST-DUR = 1
N-VOICES = ---
RATIO = 1
param1 [x] FUND
param2 [y] N-VOICES
Descriptor: mean-f...

Figure 14*: 3D pane showing how the fundamental and the number of voices are related with the mean-frequency; the mean frequency is represented as shades of grey*

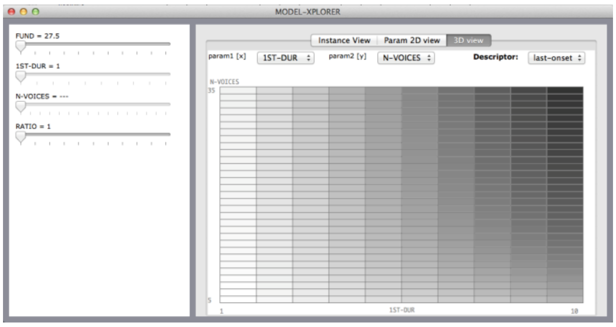

FUND = 27.5
1ST-DUR = 1
N-VOICES = ---
RATIO = 1
param1 [x] 1ST-DUR
param2 [y] N-VOICES
Descriptor: last-onset

Figure 15*: 3D pane showing how the first duration and the number of voices affect the value of the last onset*

Figure 16 shows the relation of parameters first duration and number of voices in this case: while random patterns can be identified, the overall tendency or the general profile is maintained. The same is true if we observe again the relation between number of voices and 'last onset' (Figure 17), as well as for number of voices and 'sonance' (Figure 18**)**.

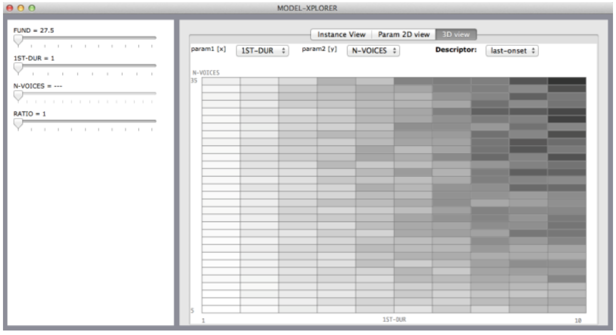

FUND = 27.5
1ST-DUR = 1
N-VOICES = ---
RATIO = 1
param1 [x] 1ST-DUR
param2 [y] N-VOICES
Descriptor: last-onset

Figure 16*: 3D pane showing the relation between the parameters 'first duration' and 'number of voices', and the value of the last onset of a set of instances, when the value of k is changing randomly each time*

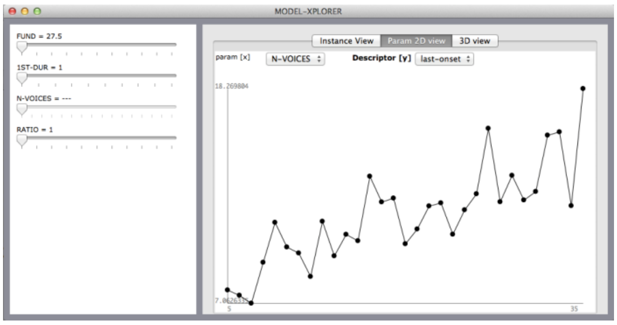

FUND = 27.5
1ST-DUR = 1
N-VOICES = ---
RATIO = 1
param [x] N-VOICES
Descriptor [y] last-onset

Figure 17*: 2D pane showing the relation between the number of voices and the value of the last onset when k is changing randomly*

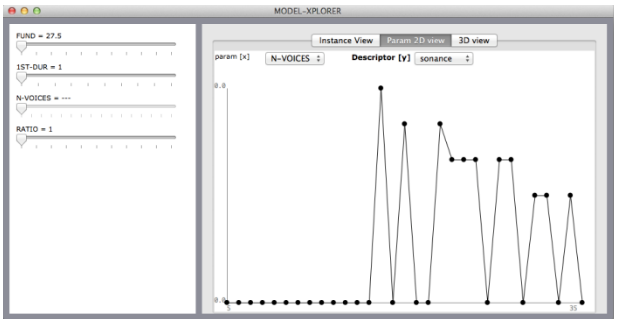

FUND = 27.5
1ST-DUR = 1
N-VOICES = ---
RATIO = 1
param [x] N-VOICES
Descriptor [y] sonance

Figure 18*: 2D pane showing the relation between the number of voices and the 'sonance' rate when the frequencies of each instance are randomly stretched*

5. PERSPECTIVES AND CONCLUSIONS

At this stage, our tool is certainly still limitative. Visualising only one measurement at a time can be restrictive, especially if such a small number of analytical descriptors are used. On the other hand, this attempt has granted us the possibility of better understanding how the model and its parameters behave. It could be used as a quick way to have a global view of the space of instances and after that select a few ones to be considered in a more detailed analysis.

We hope to develop further this tool and share it with the community to discover new and better ways of using it. For instance, we consider that the exploration of a model's various instances would greatly improve with the incorporation of optimisation or search algorithms. Instead of generating a large number of variations at once, we could run simulations until the computer program finds an instance (or a group of them) that corresponds to pre-determined features. To do so, we envisage the implementation of algorithms as simple as the *simulated annealing.* This strategy would open new applications for our models.

At the same time, with the integration of a greater number of analytical descriptors, the implementation of search algorithms would serve as a useful tool in the hands of many composers who are used to model their compositional processes or to the users of computer assisted composition software in general. It could work in an analogous way to constraint programming in computer music (Truchet & Assayag 2011).

3. REFERENCES

Assayag, G. (1993) Cao: vers la partition potentielle*. Les Cahiers de l'Ircam: La composition assistée par ordinateur* 1(3).

Bresson, J. (2007) *La synthèse sonore en composition musicale assistée par ordinateur: Modélisation et écriture du son.* Thèse de doctorat, Université Pierre et Marie Curie (Paris 6), Paris.

Bresson, J., Agon, C., & Assayag, G. (2011) OpenMusic Visual Programming Environment for Music Composition, Analysis and Research. *Proc. ACM MultiMedia 2011 (OpenSource Software Competition),* Scottsdale, AZ, USA.

De Paiva Santana, C., Bresson, J., & Andreatta, M. (2013) Modeling and Simulation: The Spectral CANON for CONLON Nancarrow by James Tenney. *Proceedings of Sound and Music Computing Conference 2013*. Stockhom, Sweden.

Guigue, D. (2010) *Sonic Object Analysis Library (SOAL*). Universidade Federal da Paraíba (UFPB), João Pessoa.

Levy, F. (2004) *Complexité grammatologique et complexité aperceptive en musique.* Thèse de doctorat, Ecoles des Hautes Etudes en Sciences Sociales, Paris.

Nattiez, J.-J. (1987) *Musicologie générale et sémiologie,* Christian Bourgeois (ed.), Paris.

Riotte, A. & Mesnage, M. (2008) *Formalismes et modèles musicaux* (2 volumes), Editions Delatour, Paris.

Rokita, L. (1996) Modèle Rythmique d'une piece pour clarinette d'Igor Stravinsky. *Journées d'Informatique Musicale* (JIM 96).

Shvets, A. & De Paiva Santana, C. (2014) Modelling Arvo Pärt's music with OpenMusic. *Proceedings of Electronic Visualisation and the Arts (EVA 2014).* London, UK, 8–10 July 2014. pp. 9–16. British Computer Society, London.

Truchet, C. & Assayag, G. (2011) *Constraint Programming in Music.* Wiley – ISTE.