Rudiments of Simulation-Based Computer-Assisted Analysis Including a Demonstration With Steve Reich's *Clapping Music*

Charles de Paiva Santana
Interdisciplinary Nucleus for Sound Communication - University of Campinas
charles@nics.unicamp.br

Abstract: *Based on previous experiments the article presents the most basic principles of a computational approach for musical analysis that through deterministic algorithms aims to reconstruct and then simulate neighboring variants (called instances) of existing musical scores. For that, adequate numerical representations are required, and their use in Computer-Aided Composition (CAC) systems are presented. Numerical sequences mapping to musical elements such as pitches, durations, and articulations may be computed or hardcoded for subsequent transformation, concatenation, and superimposition. They allow the reconstruction of the segments of a given musical score. The rhythmic pattern of Clapping Music can be modeled as a group of beats being progressively deprived of one beat, each group being separated by a rest, and the sequence concatenated with its retrograde. The sequence is subsequently transformed by the successive application of "phase shifts". A graphical interpretation of the piece is introduced using barcodes. Variations are envisaged by manipulating parameter values, each different value corresponding to a specific instance. Usually, parameters reflect compositional choices, but completely arbitrary models are possible. Such is the case of an alternative model of Clapping Music where a number is converted to binary representation and then mapped to rests and beats of eighth-notes. The manipulation of strong parameters modifies structural features of the musical score while weak parameters may only change the way the score is notated. The set of possible simulations gives rise to a space of instances. It can be analyzed through diachronic analysis, where a small group of variations is compared to the original piece, or achronic analysis, where variations are seen as single points in the space.*

Keywords*: Algorithmic composition, Computer-assisted musical analysis, Computational Musicology, Steve Reich .*

I. Introduction

In the course of the past years, I was involved in the algorithmic modeling of a couple of pieces. They included:

- *The Spectral Canon for Colon Nancarrow* by James Tenney [21, 22]
- *Spiegel im Spiegel* by Arvo Pärt [26]
- *Désordre* for piano by György Ligeti [23].

Although very different, they share a strong underlying algorithmic thinking. For this very reason, I chose those pieces as case studies of a kind of computer-assisted analysis that focus on the reconstruction and generation of neighboring variants.

In that I was inspired by the work of French researchers André Riotte and Marcel Mesnage [18, 19]. They are known to have developed a musical analysis based on formal models of compositions from the repertoire [1]. Before them, the usual was to simulate a given style, not a single piece, through the determination of probabilities and the conceptual help of Shannon's *Information Theory* [25]. Following the path opened by Fred Brooks [5], Lejaren Hiller [12] and their compatriot Pierre Barbaud [3], they used computer simulations to validate deterministic models instead. However, the possibility of using the same models to simulate variations was only suggested by Riotte and Mesnage. Figure 1 shows a comparison between the approaches.

In this article, I intend to expose some of the principles that guided me in the elaboration of such computer models. Naturally, the following ideas are still in development, and I expect them to be further expanded in future works. For such a short text, I selected what is most relevant and imperative to understand the general approach.

In the course of the following explanations I will refer to other published experiments but will include the modeling of Steve Reich's *Clapping Music* that was conceived for a pedagogical purpose [2].

Nico Schüler [24] rightly says that musical analysis, and especially when computer-assisted, is often taught and practiced with few or any references to the used methods. This article is also my first answer to that.

II. Theoretical Premise

It was demonstrated that existing musical scores could be rebuilt or generated using computer models and it was also suggested that neighboring variants could 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 neighboring variants obtained by a different set of parameters.[3] [20].

Expanding on the above definition, we can postulate that *a musical score is one single occurrence of a system's particular configuration.* Neighboring variants are then envisaged by modeling the behavior of such systems and manipulating its parameters values.

The variants are called the *instances* of the piece. Being similar or unlike, the instances are ontologically related. Their study should result in further knowledge about the musical work's inherent attributes and open for new analytic and creative possibilities.

The model is understood *as a computer program that allows the reconstruction of the musical score or some of its specific aspects.* The model inputs are parameter values, and its output is symbolic musical data. 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 on the preliminary musical analysis of the chosen piece.

[1]Those included the *Variations for piano, op. 27* by Anton Webern, The *Two-Part Invention No. 1* by J.-S. Bach, and the *Troisième Regard sur L'Enfant Jésus* by Olivier Messiaen.

[2]This text is however intentionally devoid of programming examples. For a complete (Common LISP) implementation of Clapping Music, allowing for the reconstruction of the original piece and generation of variants refer to https://github.com/charlesdepaiva/Clapping-Music

[3][...] modéliser la partition, pour nous, c'est modéliser le processus de composition par un algorithme capable de reproduire, soit la partition, soit des variantes voisines obtenues par un nouveau jeu de paramètres.

The modeling is oriented mostly towards 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*) [17, 22].

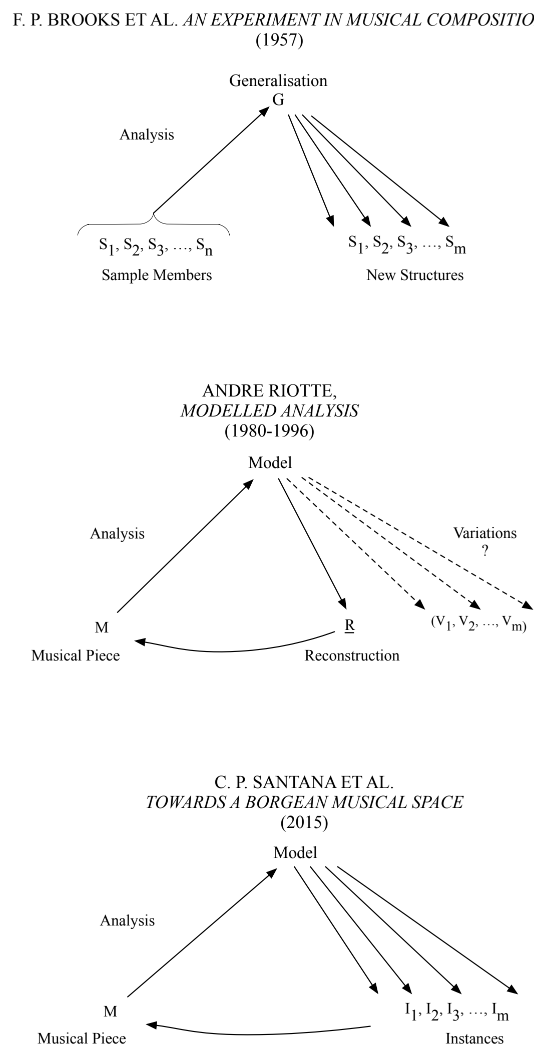


Figure 1: *Diagrammatic representation of three different approaches for modeling and simulation of musical scores. Based on Fred Brooks* An experiment in musical composition *[5]*

To implement the model and run simulations the use of a Computer-Assisted Composition environment (CAC) is of great help. Those include *OpenMusic* [4], *PWGL* [15], and *Common Music / Grace* [28, 29, 30] environments. Those offer a large set of ready-to-use *functions* performing these categories of tasks and where new routines can be constructed from them. The output of the model can be heard as an audio or MIDI file, seen as a musical score and fine-tune edited (see Figure 2).

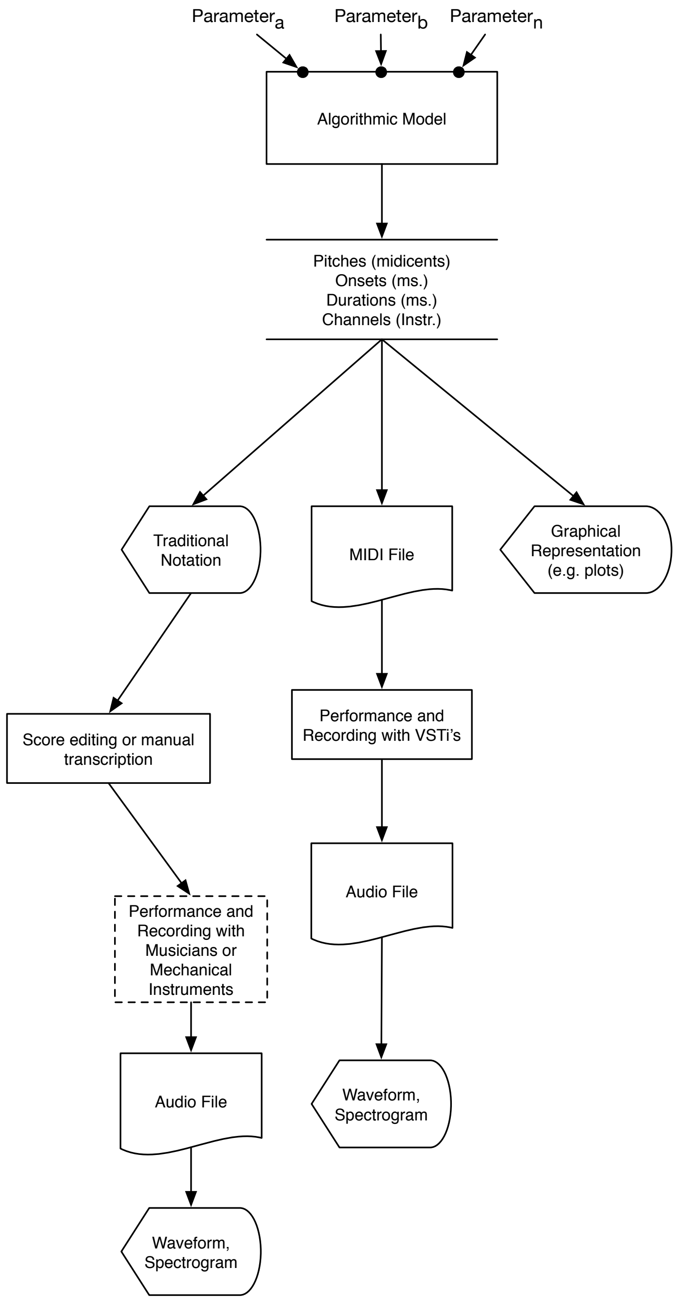


Figure 2: *Flowchart illustrating the symbolic data output by the algorithmic model being explored through different digital formats for representing music.*

III. Numerical Representations

The fundamental step of any computer-assisted analysis is the specification of adequate numerical representation for the musical structures [8]. The musical score is a representation of sound phenomena. The score is a space that suggests and privileges certain operations as a result of its bidimensional aspect. Space and time operations, as transpositions, inversions and retrogradations result from the possibilities offered by the representation itself. In this sense, the numerical representation of (notated) musical structures is a representation of a representation, and the computer model of a score is a model of a model. The numerical representation is the mediation between musical score and computer.

The numerical representation of musical structures refers us back to French composer Olivier Messiaen (1912–1992). With the piece *Modes de valeurs et d'intensités* (1949), he not only laid the foundations of *total serialism*, but also introduced a detailed mapping of different musical dimensions to integer numbers (see Figure 3). This representation allowed Messiaen to operate space and time transformations, used in polyphonic writing (and by the twelve-tone technique), not only upon pitches and durations but also intensities and articulations.

In fact, pitch and durations, because of cultural and historical developments, will be more suitable to this kind of representation and, consequently, to calculations upon them. As we know, most of the Western musical tradition, in which fall our case studies, privileged those two dimensions. On the other hand, intensities, articulations and especially timbre, can be seen as more challenging as they were not explored as much by traditional western music theory. That probably comes from the fact that, from the perspective of performance and perception, those dimensions pose some challenges, although the issue may be better handled in the electroacoustic domain.

Be as it may, one could argue that simulation-based analysis is more suitable to compositional practices that focus chiefly on the pitch and durational dimensions. Additionally, compositional practices where an underlying algorithmic thought already exists, as in some serial pieces and most of the works written by Olivier Messiaen, James Tenney, Arvo Pärt and Steve Reich, among others, may constitute a *corpus* more fitted to be studied by the modeling approach presented here.

Pitch can be represented in hertz, savarts, MIDI number and so on. The MIDI standard assigns an integer number to each key of a standard keyboard (C4=60, C#=61, D=62 . . .). As the MIDI system was not conceived for microtonal music, we can use instead the *midicent* standard, which is the MIDI number multiplied by one hundred (C4 = 6000). In the *midicent* system, one semitone is equal to 100 and one octave is equal to 1200 midicents (see Figure 4). It is used in CAC environments such as OpenMusic [4] and PGWL [15]. In some cases, as in the model of James Tenney's piece [21] and most of *spectral music*, the representation in hertz is needed for some calculations.

One way to represent durations and rhythms is to map the traditional rhythmic figures to a fractional representation. Such representation is already used to formulate measure signatures. For example, 1/8 (or just 8) refers to the eighth-note, while 1/4 refers to the quarter-note and so forth. To represent processes not based on rhythm figures (as in proportional notation) the concept of onset is used. It represents the moment in time (here expressed in milliseconds) where a note is attacked or an event is started. The onset information is complemented by the determination of the same event's duration (see Figure 5).

For dynamics, we use what is already specified by the MIDI standard, a range from 0 (very soft or silence) to 127 (very loud). It can be mapped to scales of different steps to represent music score marks, such as *piano*, *forte*, and so on. For articulations such as *legato*, *non legato* and *staccato*, which have a durational nature, we may think of a scale that goes from 0, a very short interpretation of

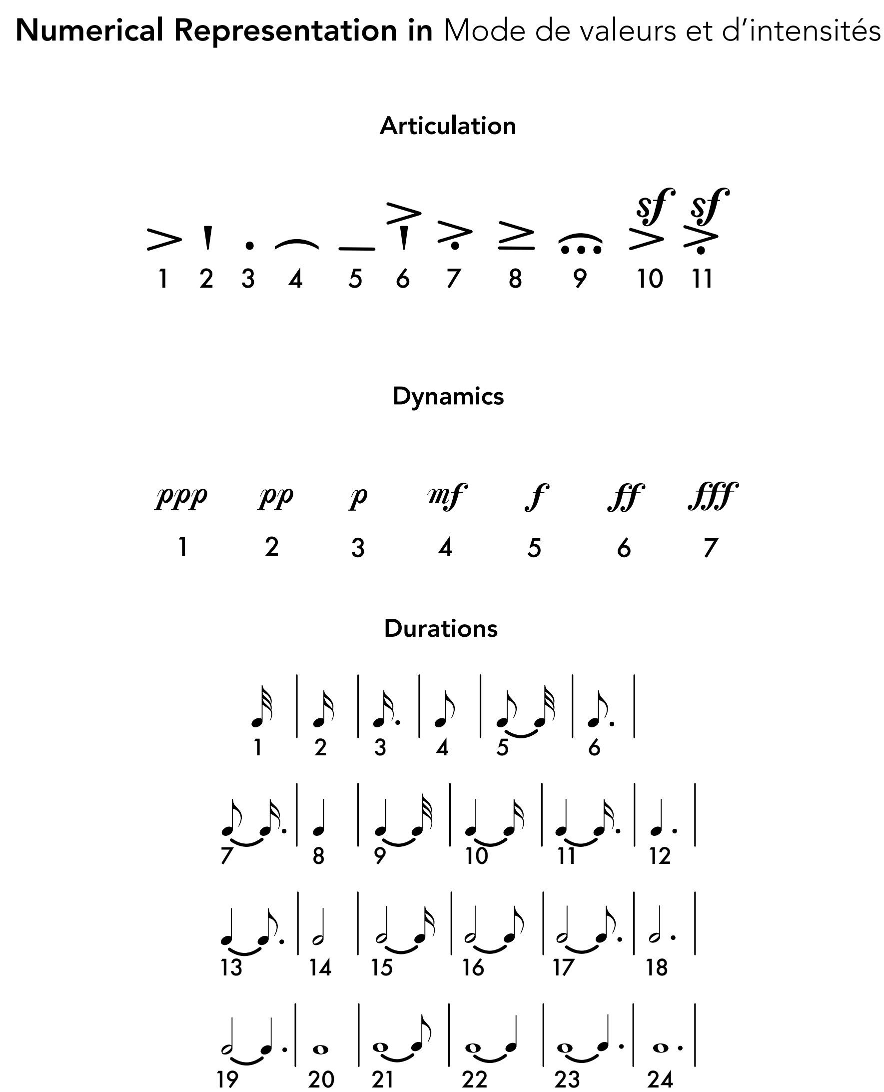


Figure 3: *Numeric representations in* Modes de valuers et d'intensités *(1949) by Olivier Messiaen.*

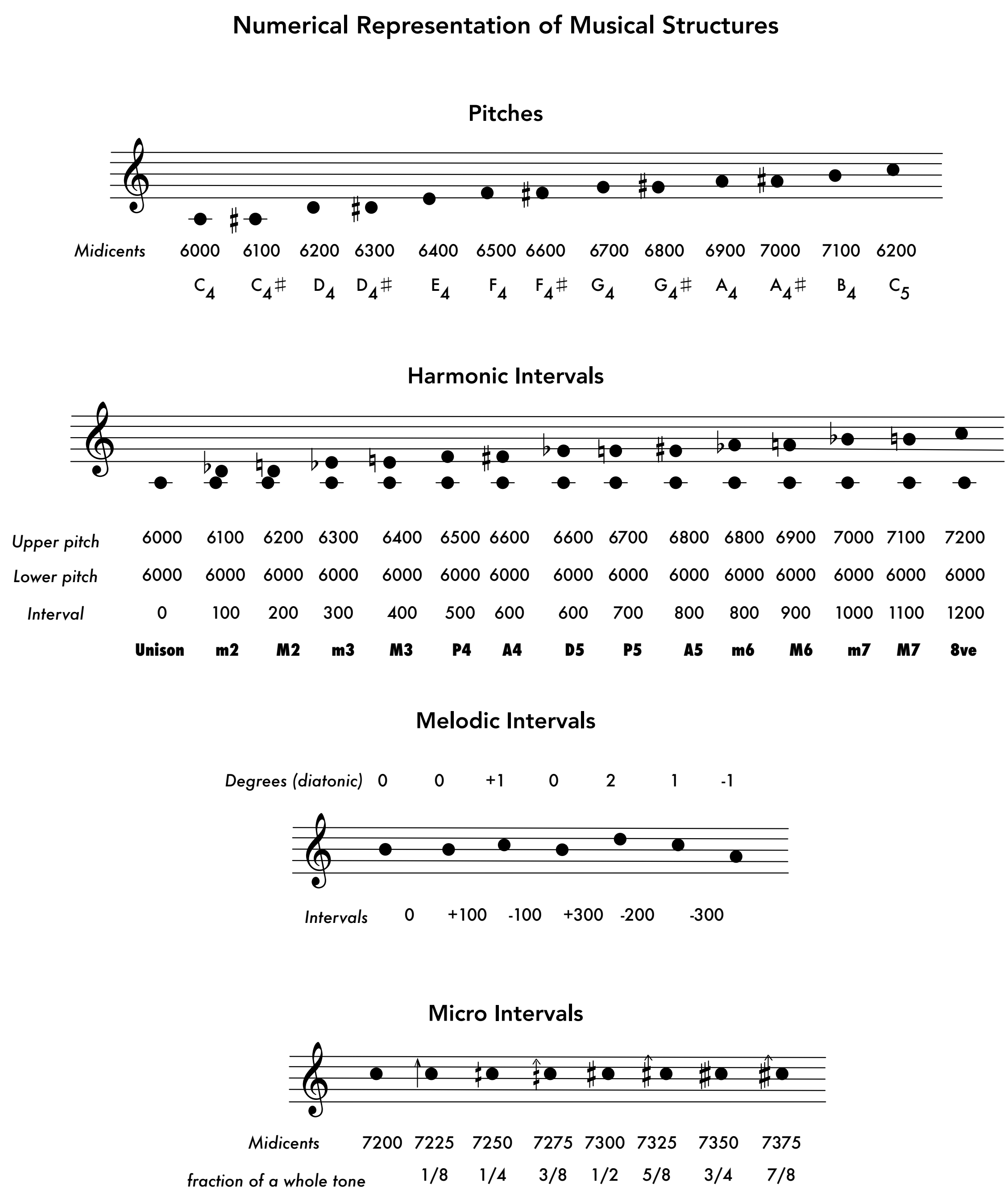


Figure 4: *Numerical representation of pitches as they are used in CAC systems such as* OpenMusic *and* PWGL*. The* midicent *number is the MIDI number multiplied by 100, allowing the use of steps smaller than the semitone.* Common Music / Grace *however uses a decimal MIDI number representation for micro intervals.*

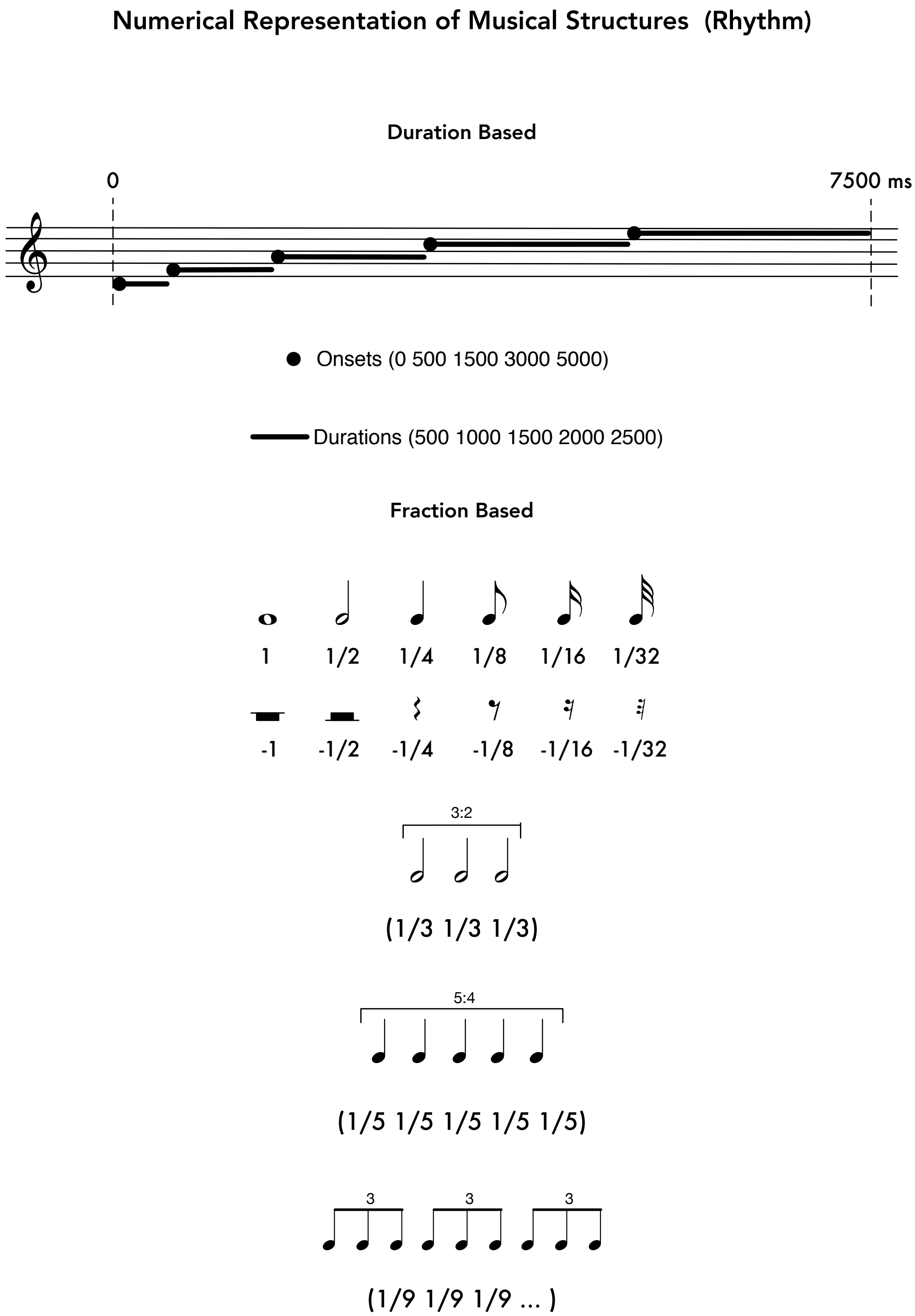


Figure 5: *Numeric Representation of rhythm and durations. Onsets and durations are represented in milliseconds.*

the 'written' duration as in *staccatissimo*, to 100, as in *legato*, where the duration, if needed, may be prolonged to connect the notes one after another (see Figure 6).

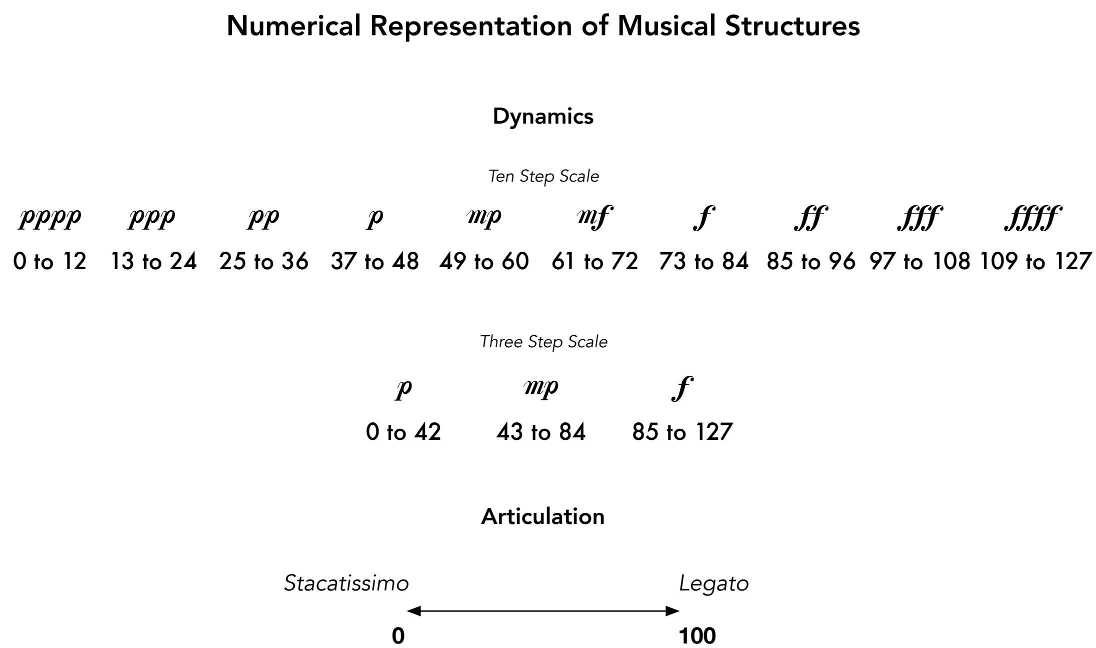


Figure 6: *Numeric Representation: dynamics (from MIDI standard) and articulation (as used in* OpenMusic*).*

IV. Generation of Numerical Sequences

Very often, modeling scores and compositional processes involve the segmentation of sequences of symbolic data, like pitches, durations, etc. More precisely, it is interested in the formalisms that correlate those segments and generate them. For instance, a model can describe the rules from which a sequence, or pattern, is *transformed*, *concatenated* and *superimposed* to reconstruct a given musical score or excerpt. Such is the case when a melody is consistently repeated (concatenation) and transposed (transformation), as in Ligeti's *Désordre* [30, 14]. That is also the case when a rhythmic pattern is repeated, rotated (transformation) and played in a texture of two voices (superimposed), as it happens in *Clapping Music*.

The model can treat a sequence, or segment of, monolithically, that is to say, consider it as a "given series" and "hardcode" it on the implementation. Or the model can describe how its most basic patterns can be computed and generated. In both cases, such endeavor often overlaps with some domains in discrete mathematics and computer science, especially those of *Formal languages* and *Automata theory* [19, 6]. Those fields are interested in the study of numerical sequences and the different, often abstract, machines that can compute them [31, 1, 9, 16].

In the modeling of James Tenney's *Spectral Canon for Conlon Nancarrow* there is almost any transformations, and the entire sequence of durations is modeled by a single equation (the sequence is then concatenated with its reverse and superimposed) [21]. On the other hand, the modeling of a twelve-tone piece may be interested only in the row's transformations and not in how the twelve-tone series itself can be computed.

In short, the modeling process means finding algorithms that can reconstruct and transform numerical sequences which, in turn, are mapped to different musical elements as pitches, durations, dynamics and so on.

Another aspect that can be part of the modeling process is the *generation of music materials*. It is considered as musical material the set of chords, scales, articulations and so on. Broadly speaking it involves the study of "sets", where the sequential or temporal aspects are not considered yet. For instance, it concerns the generation of pitches from the harmonic series for the *Spectral Canon for Colon Nancarrow* [21], or the generation of pitches from the concept of "combinatoric tonality" in György Ligeti's *Désordre* [27, p. 8]. According to the categorization presented by Chemilier [7], after Xenakis, the generation of music materials is part, essentially, of what can be called the *outside time domain*.

Now, to demonstrate the modeling of a rhythmic sequence I will refer to the pattern of *Clapping Music* (also used in *Music for Pieces of Wood*). The pattern can be interpreted as a borrowing from Subsaharan African music or an exercise drum pattern. In any case, we could see it as a result of an algorithmic process where (1) a group of eighth notes is progressively deprived of one beat, (2) each group is separated by a rest, (3) and the current sequence is followed by its retrograde. Figure 7, shows its analysis. This sequence is repeated, transformed, and arranged in a particular form. Figure 8 shows a graphical interpretation of the piece.

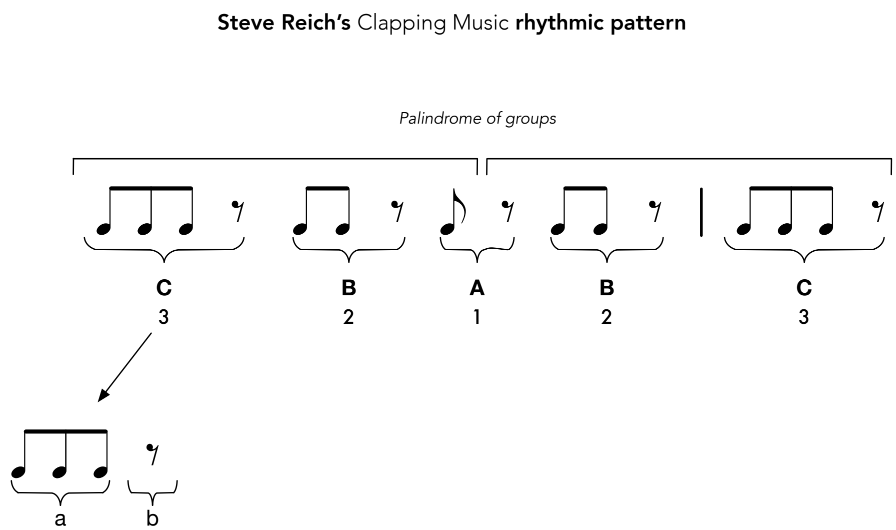


Figure 7: *Analytic decomposition of Steve Reich's* Clapping Music *rhythmic pattern.*

V. Transformations and Operations

Once established the representations, materials, and segmentation, the next step is the determination of the transformations that operate on those numerical structures. Some of the most recognizable transformations of Western music literature can be implemented by simple arithmetic, for example, transpositions can be made by adding to a sequence of numbers (representing a melody or chord) a particular interval. The intervals between pitches can be implemented as the

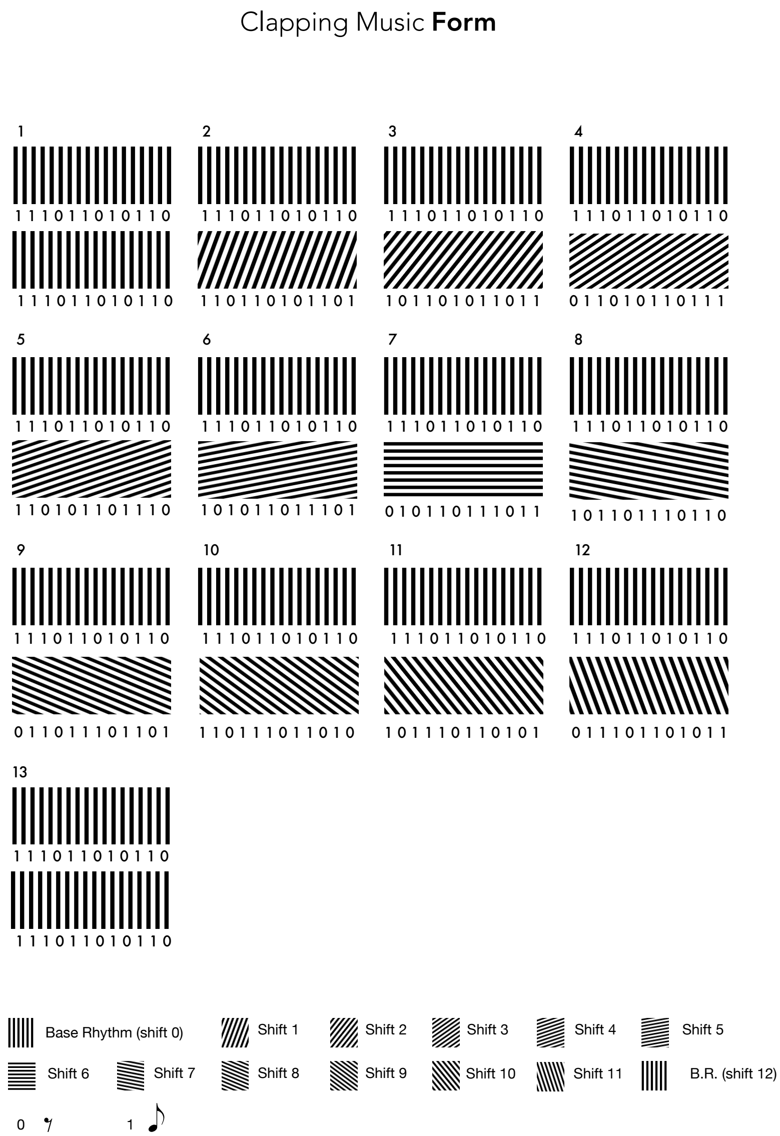


Figure 8: *Musical form of Steve Reich's* Clapping Music*. The rhythmic pattern is numerically represented by zeros (rests) and ones (eighth-notes). While the first voice only repeats the pattern, the second voice successively* transforms *it through the application of* phase shifts *(cf. Figure 10). The repetitions and shifts are also represented by the different 'barcodes'.*

subtraction of highest and lowest notes. Processes as rhythmic augmentation and diminution can be calculated as multiplications of the pattern.

Very often *special transformations* or *operations*, that is, compositional processes particular to a given composer or practice, need to be implemented. An example of special transformation is the *tintinnabulation* used by Arvo Pärt, where a melodic sequence is consistently mapped to the tonic's triad [13]. Another example of special transformation is the technique of Phase Shifting used by Steve Reich. In the specific case of *Clapping Music*, it means the consistent rotation of its rhythmic pattern (Figure 10).

The implementation of special transformations may involve the preliminary study of available literature, as the composer's texts and sketches. Nevertheless, transformations and even complete models could be made from independent, arbitrary generative processes. To demonstrate the modeling by an arbitrary process, in the Figure 9 we see *Clapping Music*'s rhythmic pattern represented through the mapping of rests and eighth-notes to zeros and ones. Then successive *phase shifts* are applied to the that binary sequence (Figure 13). We consider this process as arbitrary because it displays a weak musical thought; the manipulation of its only parameter, a decimal number that is then converted to a binary string, gives very little control of the musical output.

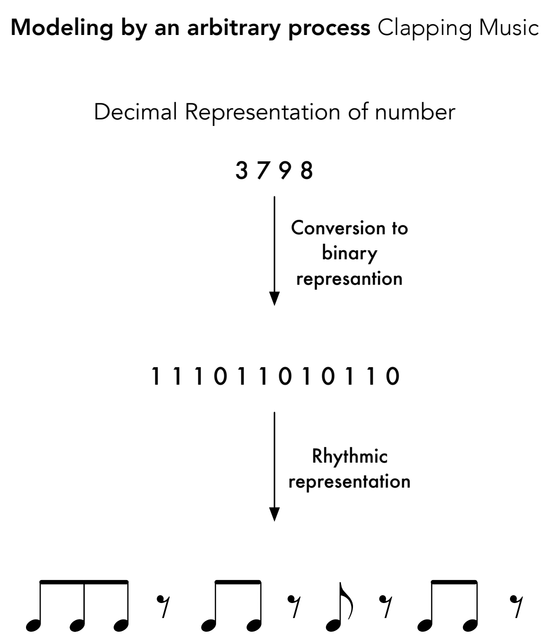
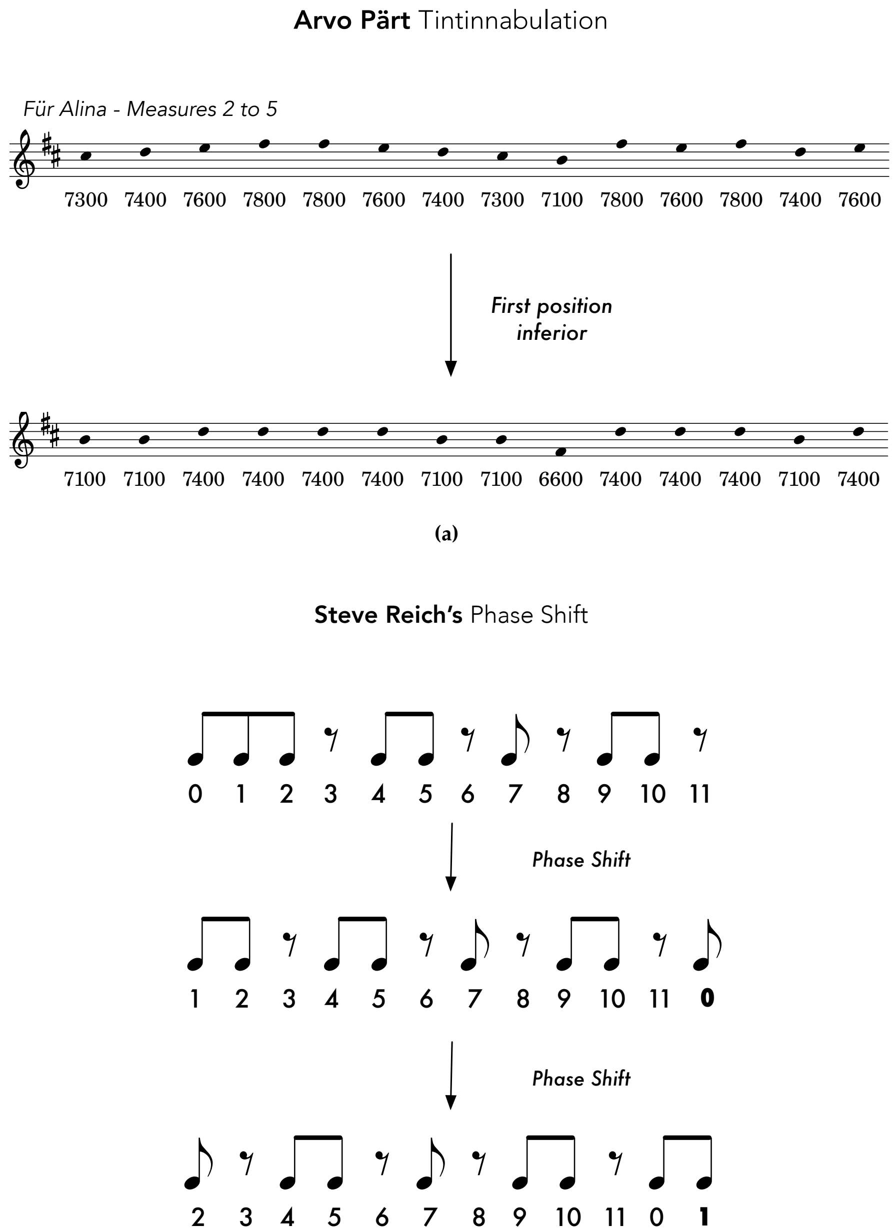


Figure 9: Clapping Music*'s rhythmic pattern. By changing the representation of a decimal number, 3798, to binary and mapping its ones and zeros to beat and rests the pattern can be reconstructed. Applying this same procedure to different decimal numbers can generate new patterns, but there is so little control of the musical output that this procedure is not so different from a random process. Compare with Figure 11.*

VI. Scope of the Model and Its Parameters

In most cases, a model will reproduce a partial section of a piece or one of its specific dimensions. For instance, Rokita [20] modeled only the rhythmic aspects of the first of the *Three Pieces for Clarinet Solo* by Igor Stravinsky. Chemillier [7] modeled only a few measures of Ligeti's *Melodien*.



(b)

Figure 10: *Examples of special* transformations *or* operations. *In (a) each pitch of a melodic sequence is mapped to the closest one of the tonic's triad. This technique is used in pieces such as* Für Alina *and* Spiegel im Spiegel. *In (b) a rhythmic pattern (from* Clapping Music*) is consistently rotated. When those rotations are superimposed with the original underlying "phase patterns" emerge.*

Occasionally one can conceive an exhaustive model for a piece in all its extension and most of its dimensions (pitch, rhythm, dynamics, etc.), like the model for James Tenney's *Spectral Canon* [21] or most of Riotte and Mesnage examples [19]. Alternatively, the modeling can be concerned uniquely with a particular technique or set of them.

The conception of parameters is an essential step; they control the model's behavior and are responsible for generating neighboring variants.

Usually, the parameters reflect a compositional choice. It can concern aspects like instrumentation, tonality, mode, tunning, and so forth. For example, in a canonic piece, one of the parameters could be the starting point for each voice or how many canonic voices should be written (see [21]). The creation of parameters for a model may depend on the creativity and purposes of the researcher. The determination of the parameters will significantly influence the implementation process and the model's capacity for generating more or less neighboring variants. To exemplify the elaboration of parameters, Figure 11, shows the modeling of *Clapping Music* rhythmic pattern this time using an essentially more musical procedure. Aspects as the number of beats inside rhythmic groups, the inclusion of gaps between groups and the notational figures (in this case the beat-unit) can be considered as parameters (compare with Figure 9).

Parameters may have a strong or weak effect on the simulations of the model. *Strong parameters* change structural features of the modeled piece while *weak parameters* may only transpose the whole structure maintaining the same structural relations between its elements or only change the way the score is notated. In the illustration showed in Figure 11 parameters `length` and `step` are strong parameters that can the structure of the rhythmic pattern. On the other hand, parameter `figure` only changes the speed of the pattern or the way it is notated. In the model of the *Spectral Canon* [21]), the parameter `fundamental` is weak because its effect is only a transposition of the whole pitch structure; its internal relationships are not modified when its value changes.

In the case of a deterministic model, every single parameter value will correspond to a particular output, to a specific *instance* (cf. Figure 12). When we plug the same values to the parameters in successive simulations, we can be sure that the instances produced will also be the same. By using deterministic algorithms we can adjust the model according to the simulation's results, i.e., we can modify the model to shape a determined space of instances.

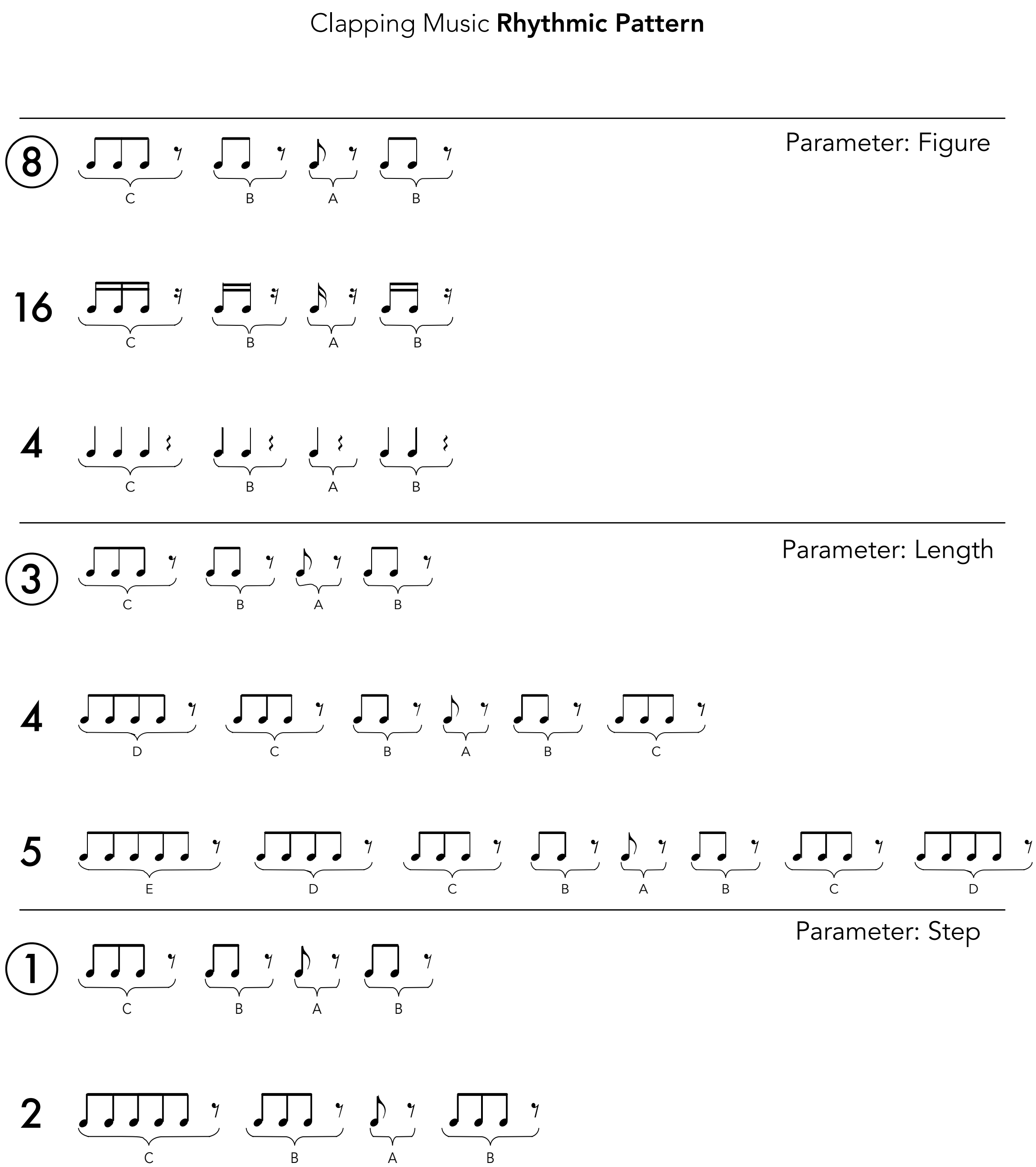


Figure 11: *Conceptualization of parameters for the rhythmic pattern used in* Clapping Music*. The parameter* `figure` *changes the notation and speed of the pattern, 8 meaning an eighth-note. Parameter* `length` *controls the length and the number of beats in a rhythmic group while the parameter* `step` *can insert gaps between the rhythmic groups of the pattern.*

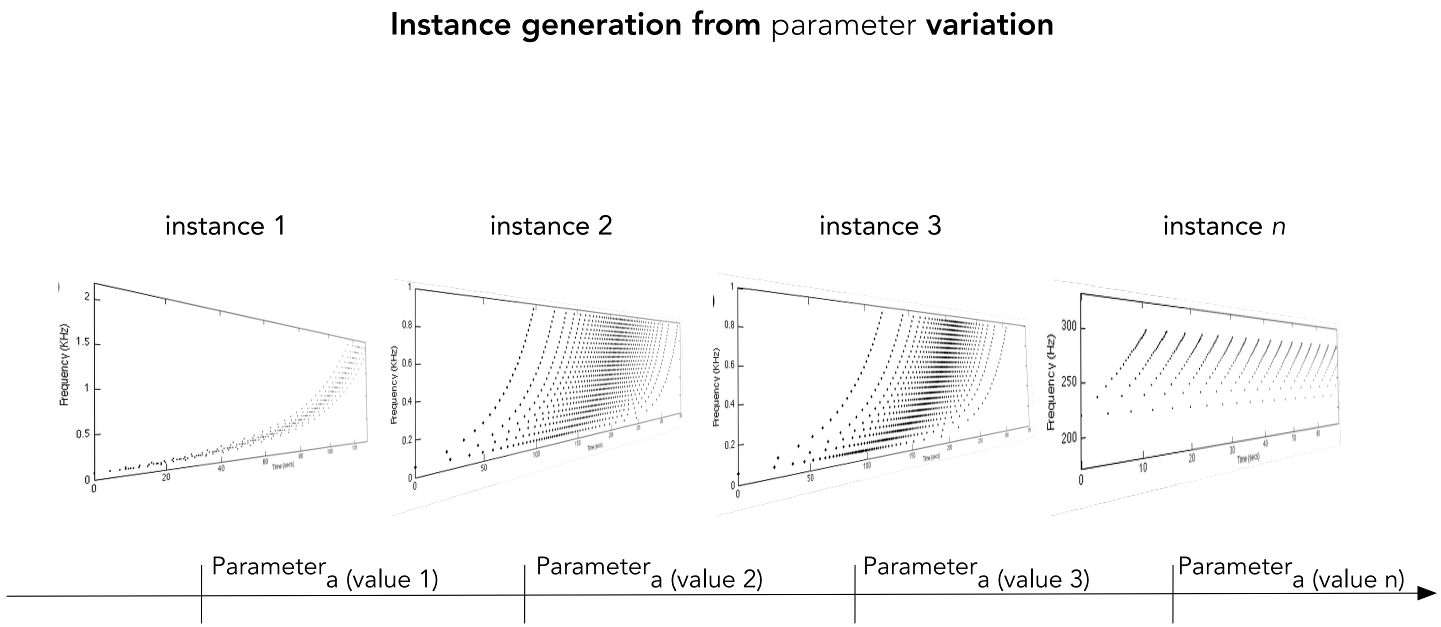


Figure 12: *Conceptualization of neighboring variants generation* (instances) *by plugging different values to a parameter. Each value corresponds to a specific* instance. *The rectangular boxes representing the instances are piano rolls (time x frequency).*

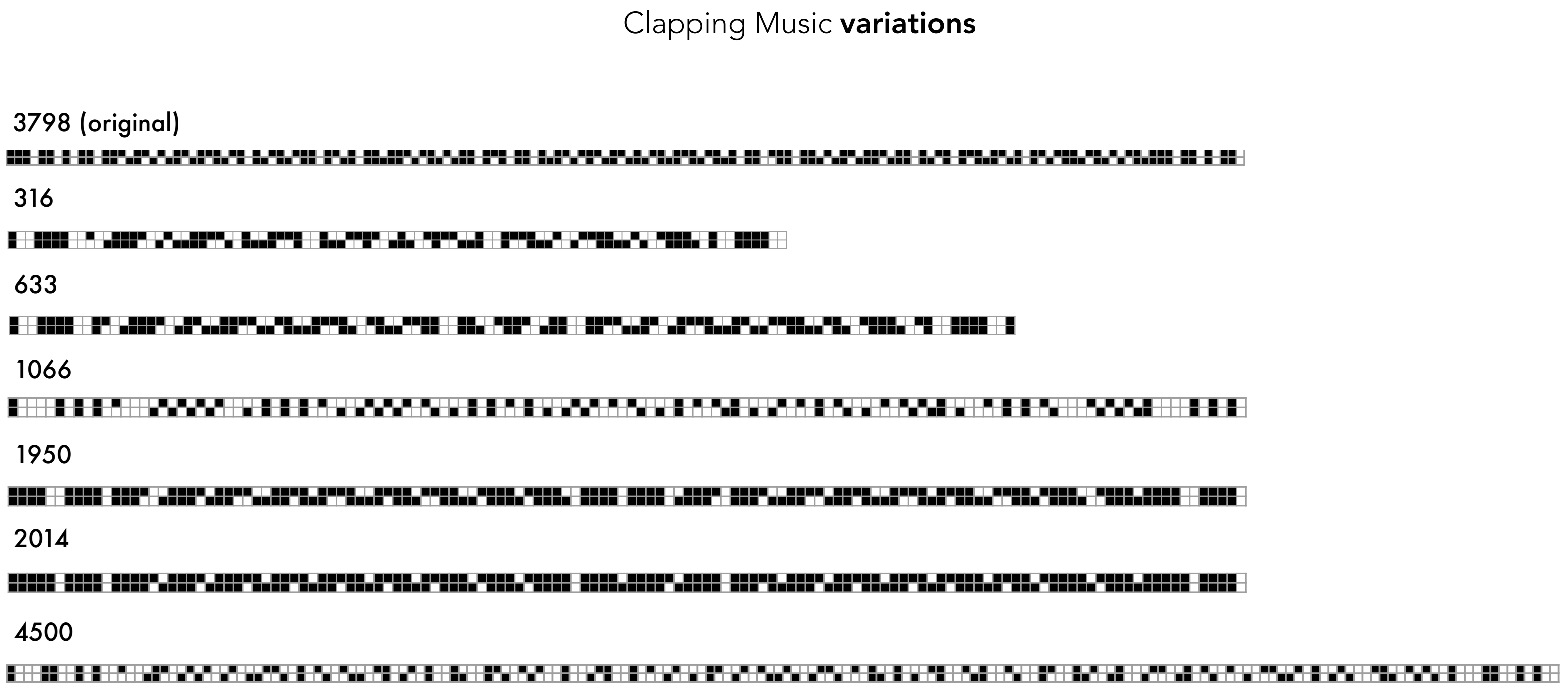


Figure 13: *Different variations obtained from simulating* Clapping Music *from the arbitrary process shown in Figure 9: Each given decimal number is converted to binary and then mapped to beats and rests. The resulting patterns are phase shifted and repeated accordingly to build the original piece and the variations. White boxes represents rests and black boxes eighth-notes. The* ritornellos *are not included in neither the reconstruction or variations.*

The fewer parameters a model has, the higher is its explanatory potential, as a more comprehensive systematization will be required to connect all the generative process with fewer variables. On the other hand, the more parameters a model has, the greater is its potential to generate different instances, thus serving to more creative (compositional) or speculative purposes [2] (Figure 14).

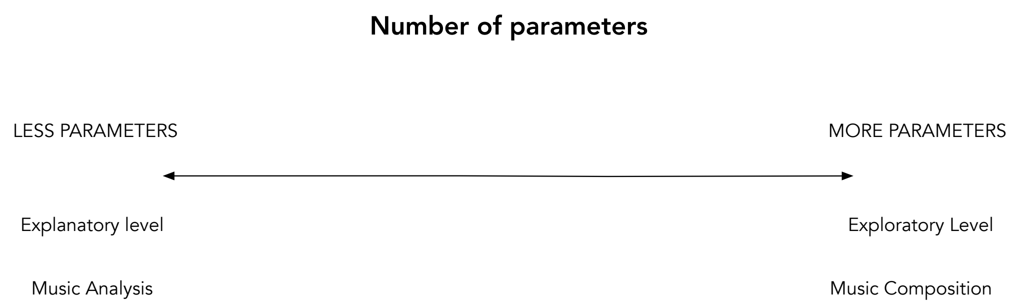


Figure 14: *The less parameters the model has, higher is its explanatory potential. The more parameters a model has, higher is its potential to generate different instances.*

Different models can be conceived for one single piece. The conception of a model depends on the hypothesis and purposes of the researcher. In the same way, different implementations are possible for one single model. Also, as with any computer program, an implementation can have several versions and be developed in several forms (see Fig 15). During the modeling and implementation process, adaptations can continuously be made to better adjust the model's output to the numerical representation of piece being modeled.

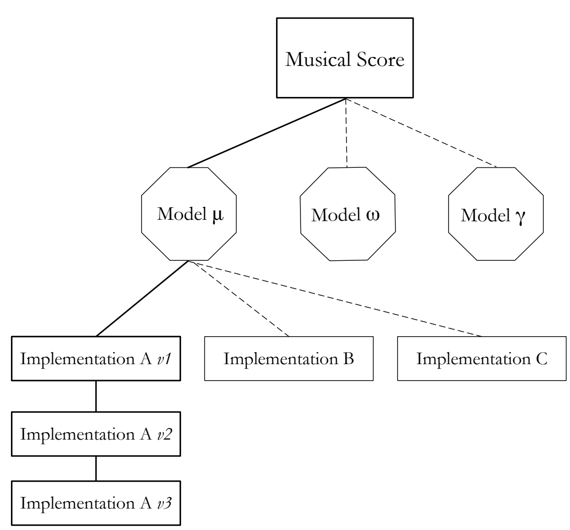


Figure 15: *Modeling and implementation process. Different models can be conceived for one single musical score. A single model, in turn, can be implemented by different implementations.*

In the case of the model presented in [21], we first developed a model with only a few parameters, to test compositional decisions, to understand the underlying principles of the composition, and then we added more parameters as a way to explore the potential of the model itself and not only the features of the original composition. Those type of additional parameter were introduced as *Extended Parameters* Another way to see this process is to think as a composer who first analyses a piece to learn about the aspects he is interested in but wants, in a later moment, to emulate them in a new composition, adding to the model his particular procedures.

VII. The Space of Instances

The set of possible variations for each parameter gives origin to a *space of instances*. One way to analyze this space is to simulate the effect of a specific parameter on the musical features of a sample of different instances. This analysis can be done in two ways, the *diachronic* and *achronic* analysis (see Figure 16), concepts borrowed from the *Sonic Object Analysis Library* (SOAL) written for *OpenMusic* and conceived in a different context [11].

In the *diachronic analysis*, a small sample of different instances is selected, and some features of the musical structures are analyzed. Those features are then compared to the original piece, always considering their evolution through time, throughout its extension. One single instance can be represented as a series of points in a specific time span, as is the case of a *piano roll*. It can be seen as a way to test hypothesis on a limited number of specific instances, where some of its precise details will be taken into account. Diachronic analysis is seen in [22, p. 75] (cf. Figures 5–8).

On the other hand, this method may not be the most appropriate to evaluate the dynamics of a larger sample of the space of instances. The *achronic analysis* supplies this need (see Figure 17). In this analysis, each instance is reduced to one measurement (or descriptor) such as total duration, shortest duration, *ambitus*, mean pitch, and so forth. Individual instances are seen as a single point in a determined space, where the instances' internal time is considered only implicitly. Diachronic analysis is seen in [22, pp. 77–78] (cf. Figures 10–16).

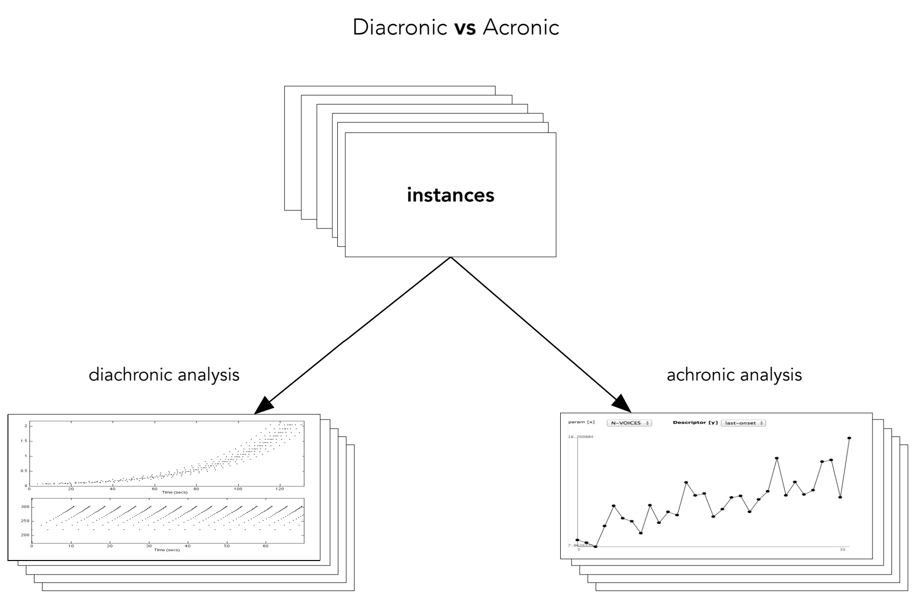


Figure 16: *In Diacronic analysis, one single instance can be represented as a series of points in a specific time span, as in the case of a piano roll. In Acronic analysis, one single instance is seen as a single point in a determined space, where the instances' internal time is considered only implicitly.*

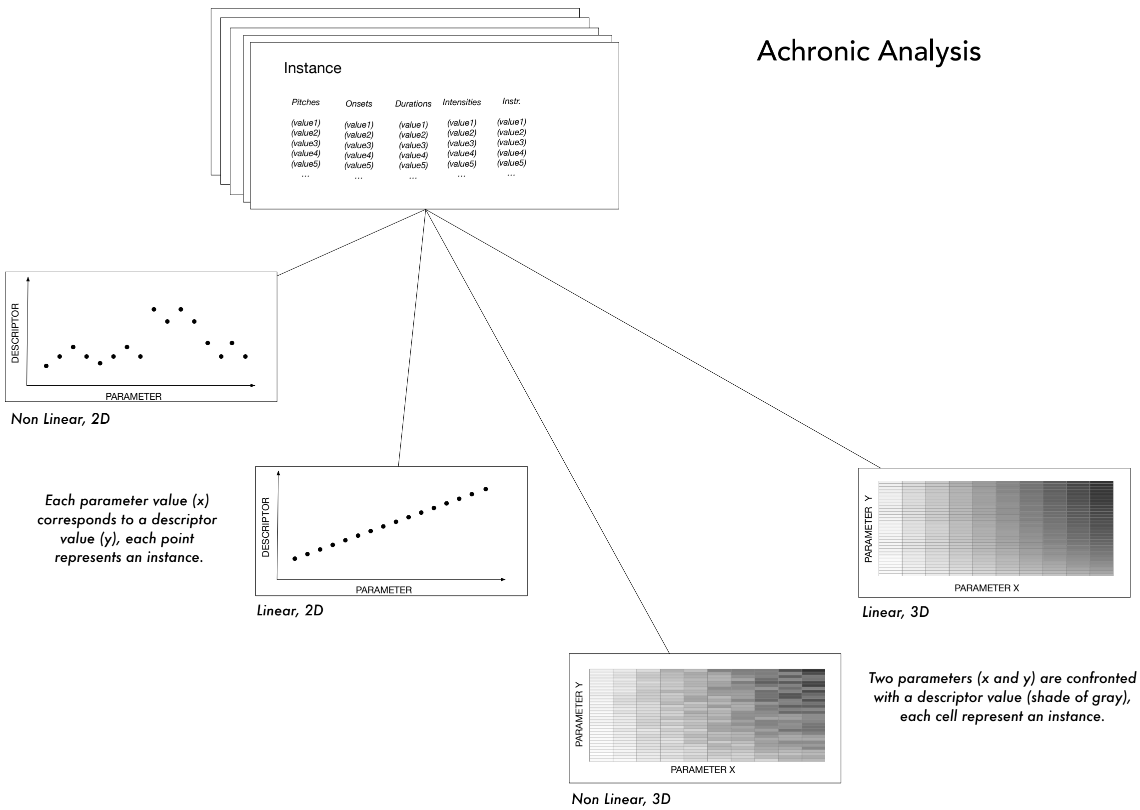


Figure 17: *Some strategies to visualize the space of instances with achronic analysis. The 2D visualization is a Cartesian plane where a subset of values of a given parameter is plotted against a measurement from the resulting instances (such as total duration, or the mean pitch, or the smallest durations, etc). In the 3D visualization subsets of values from two distinct parameters are plotted in a heat-map where a shade of gray represents a measurement from the resulting instances.*

Through *achronic* and *diachronic* analysis, we observed that, as a result of the utilization of deterministic algorithms, there might be a consistent, very linear behavior in such spaces of *instances*. That is to say, changes in the value of parameters lead to proportional responses of the simulation results. For example, if one of the parameters is the first duration, increasing it will make the total duration of the resulting *instance* proportionally bigger. If the parameter is the fundamental frequency (as in the case of a *spectral composition*), we may also expect that increasing it will also increase every frequency (pitch) of the *instances* produced by the simulation.

On the one hand, the linearity observed in the space of instances can be interpreted as a sort of validation, an element of coherence, of the model's behavior and therefore would be a desirable feature from a more musicological and pedagogical perspective. It can be used as a way to further understand specific compositional processes and explore the consequences of particular decisions. On the other hand, from a more creative, speculative attitude such a space of instances could appear as too homogeneous and predictable for some purposes.

We tried to conceive some strategies to break this linearity and introduce more heterogeneity on the space of instances. One method was to generate *perturbations* in a structural element of the model, namely a variable, by multiplying it by a pseudo-random, controlled, number. In this way, the higher the perturbation, the higher is the probability of unexpected simulation results [22, pp. 77–78].

VIII. Final Considerations

The basics of a simulation-based, computer-assisted approach for musical analysis were introduced in this paper. Some of those principles are common to other methodologies, and some are original. The case studies are not too many and come from a somewhat restricted compositional context. Nevertheless, I hope that some of the original concepts here presented will form the basis to bolder and more encompassing experiments.

The modeling of musical scores, with the explicit goal of generating neighboring variants, perhaps is one step further into the direction of a more *creative analysis*, like the one forecast and advocated decades ago by both Luciano Berio and Pierre Boulez. On the other hand, there are indeed still too much to be explored and to be learned from it. The field is open, and there is an infinity of solutions that could be tried to examine the potential for creating new musical forms from a given model.

Unsurprisingly, presenting such a method also poses some challenges. One of them is how to describe the algorithms or model implementations comprehensively without turning it a too tedious, tiresome endeavor. Additionally, the description and evaluation of several *instances* can be too burdensome, as the discussion around just one single neighboring variant can be not easily exhaustible. The automatic estimation of symbolical and psychoacoustic measurements may help in this exploration. Also, much more is needed to support better the visualization of the *space of instances* that a model can produce.

To paraphrase pioneer Stanley Gill [10], the musical results produced by the different models we worked so far "touched" me quite enough. I believe that the favorable results come from having implemented *expressive* parameters that give greater control of the musical result. For instance, some variations of Tenney's *Spectral Canon* or Steve Reich's *Clapping Music* seems to be authentic enough to be conceivably appreciated on its own. For that, the neighboring variants need only to be coherent, to retain a sense of form and achievement.

Finally, I feel that to use the computer to automate the process of discovering and reconstructing patterns, in the context of computational musicology, is not what is more important. It is more desirable to reveal new meanings to already known musical phenomena, to favor *polysemy*, and avoid the trap of reducing musical expressiveness to a fixed, inflexible algorithm.

IX. Acknowledgments

This paper put together research developed at the *Interdisciplinary Nucleus for Sound Studies* (NICS) and the *Institut de Recherche et Coordination Acoustique/Musique* (IRCAM) as part of a doctoral thesis. It was initially funded by the *Institut de Recherche pour le Development* (IRD) and later by the Sao Paulo Research Foundation (FAPESP), grant 2014/08525–8.

References

[1] Allouche, J.P., and Shallit, J. 2003. *Automatic Sequences: Theory, Applications, Generalizations*. Cambridge University Press.

[2] Assayag, Gérard. 1993. CAO: vers la partition potentielle. *Cahiers de l'Ircam n 3, Recherche musicale: La composition assistée par ordinateur*, p. 1.

[3] Barbaud, Pierre. 1960. Musique algorithmique. *Esprit (1940-)*, v.280, n.1, pp. 92–97.

[4] Bresson, Jean, Agon, Carlos, and Assayag, Gérard. 2011. OpenMusic: visual programming environment for music composition, analysis and research. In *Proceedings of the 19th ACM international conference on Multimedia*, pp. 743–746. ACM.

[5] Brooks, Frederick P, Hopkins, A L, Neumann, Peter G Neumann, and Wright, W V .1957. An experiment in musical composition. *IRE Transactions on Electronic Computers*, v.3, pp. 175–182.

[6] Chemillier, Marc. 1988. *Toward a theory of formal musical languages*. Ann Arbor, MI: Michigan Publishing, University of Michigan Library.

[7] Chemillier, Marc. 1995. Analysis and Computer Reconstruction of a Musical Fragment of György Ligeti's Melodien. *Muzica*, v.6, n.2, pp. 34–48.

[8] Dannenberg, Roger B. 1993. Music representation issues, techniques, and systems. *Computer Music Journal*, v.17, n.3, pp. 20–30.

[9] Flajolet, P., and Sedgewick, R. 2009. *Analytic Combinatorics*. Cambridge University Press.

[10] Gill, Stanley. 1963. A Technique for the Composition of Music in a Computer. *The Computer Journal*, v.6, n.2, pp. 129–133.

[11] Guigue, Didier. 2010. Une Etude 'pour les sonorités opposées': Principes méthodologiques d'une analyse orientée objets de la musique du XXe siècle.

[12] Hiller, Lejaren Arthur and Isaacson, Leonard M. 1979. *Experimental Music; Composition with an electronic computer*. Greenwood Publishing Group Inc., Westport.

[13] Hillier, P. 1997. *Arvo Pärt*. Clarendon Press.

[14] Kinzler, Hartmuth. 1991. György Ligeti: Decision and automatism in 'Désordre', 1re étude, premier livre. *Journal of New Music Research*, v.20, n.2, pp. 89–124.

[15] Laurson, Mikael, Kuuskankare, Mika, and Norilo, Vesa. 2009. An overview of PWGL, a visual programming environment for music. *Computer Music Journal*, v.33,, n.1, pp. 19–31.

[16] Linz, P. 2016. *An Introduction to Formal Languages and Automata*. Jones & Bartlett Learning.

[17] Nattiez, Jean-Jacques. 1987. *Musicologie générale et sémiologie*. Christian Bourgois, Paris.

[18] Riotte, André and Mesnage, Marcel. 1993. Modélisation informatique de partitions, analyse et composition assistée. *Les cahiers de l'IRCAM: la composition assistée par ordinateur*.

[19] Riotte, André and Mesnage, Marcel. 2006. Formalismes et modèles musicaux. *Editions Delatour France/IRCAM*, v.2.

[20] Rokita, L. 1996. Modèle Rythmique d'une piece pour clarinette d'Igor Stravinsky. *Troisièmes Journées d'Informatique Musicale, JIM'96*, pp. 277–286.

[21] Santana, Charles de Paiva, Bresson, Jean, and Andreatta, Moreno. 2013. Modeling and simulation: The spectral canon for conlon nancarrow by James Tenney. In *Proceedings of Sound and Music Computing Conference 2013*.

[22] Santana, Charles de Paiva, Mazolli, Jônatas, Bresson, Jean, and Andreatta, Moreno. 2015. Towards a borgean musical space: an experimental interface for exploring musical models. In *Proceedings of the Conference on Electronic Visualisation and the Arts*, pp. 72–79. British Computer Society.

[23] Santana, Charles de Paiva, Freitas, Vitor de Mello, and Manzolli, Jônatas. A new algorithmic model and simulation of neighboring variants for György Ligeti's *Désordre*, first piano étude. Opus - Revista Eletrônica da Anppom. 2018 (forthcoming).

[24] Schüler, Nico. 2000. Towards a General Methodological Classification of Computer-Assisted Music Analysis. In *Proceedings of the International Computer Music Conference Proceedings (ICMC 2000)*, Berlin. ICMA.

[25] Shannon, Claude E, Weaver, Warren, and Burks, Arthur W. 1951. The mathematical theory of communication.

[26] Shvets, Anna and Santana, Charles de Paiva. 2014. Modelling Arvo Pärt's music with Openmusic. In *EVA*.

[27] Steinitz, Richard. 2011. *György Ligeti*. First Trilogy. Faber & Faber, London.

[28] Taube, Heinrich. 1991. Common music: A music composition language in common lisp and clos. *Computer Music Journal*, v.15, n.2, pp. 21–32.

[29] Taube, Heinrich. 1997. An introduction to common music. *Computer Music Journal*, v.21, n.1, pp. 29–34.

[30] Taube, Heinrich. 2013 *Notes from the Metalevel: An Introduction to Computer Composition*. Taylor & Francis, Abingdon.

[31] Wilf, H.S. 2014. *Generatingfunctionology*. Elsevier Science.