Concatenative Sound Synthesis as a Technomorphic Model in Computer-Aided Composition

Júlio Guatimosim[1], **José Henrique Padovani**[1], **Carlos Guatimosim**[1]

[1]LaPIS / CrIS – School of Music/Federal University of Minas Gerais
Av. Antônio Carlos, 6627 – 31270-901 Belo Horizonte, MG

juliogn@ufmg.br, jhp@ufmg.br, cguatimosim@alumni.usp.br

Abstract. *The text presents a process aimed at computer-aided composition for percussion instruments based on Concatenative Sound Synthesis (CSS). After the introduction, we address the concept of "technomorphism" and the influence of electroacoustic techniques in instrumental composition. The third section covers processes of instrumental sound synthesis and its development in the context of Computer-Aided Composition (CAC) and Computer-Aided Music Orchestration (CAMO). Then, we describe the general principles of Concatenative Sound Synthesis. The fifth section covers our adaptation of CSS as a technomorphic model for Computer-Aided Composition/Orchestration, employing a corpus of percussion sounds/instruments. In the final section, we discuss future developments and the mains characteristics of our implementation and strategy.*

1. Introduction

Since the remarkable technological transformation triggered by electroacoustic instruments and techniques, new instrumental composition strategies emerged [1]. At the same time, the development of digital computers led to the development of algorithmic compositional practices, which culminated in research on computer music and the development of compositional practices associated with algorithmic and computational resources.

While the first process led to what is now called *technomorphisms* in musical composition, the second process led to the development of Computer-Aided Composition (CAC) and, later, to the establishment of areas such as Music Information Retrieval (MIR) and Auditory Scene Analysis (ASA): fields that require the automatic detection of features from audio files and streams.

In the present work, we address a general review of CAC and Computer-Aided Music Orchestration (CAMO) processes and the concept of *technomorphism* itself. Then, we describe the implementation of a "technomorphic" approach to music composition, aimed at percussion instruments, based on the technique of digital synthesis called Concatenative Sound Synthesis (CSS). After presenting a general review of the concept of *technomorphism* and Instrumental Sound Synthesis processes in the context of computer-aided orchestration/composition, we outline the principles of CSS to, then, present our technomorphic implementation of this technique. In the end, we discuss the potential of the implemented process, its future consequences, and the general characteristics of the approach used in this work.

2. Technomorphism in music composition

The influence of electroacoustic techniques on instrumental music at the beginning of the second half of the 20th century is notable in the work of composers whose creative work encompassed these two practices. In this context, it is not rare the application of technical means as models or metaphors for procedures used in instrumental compositions that do not use directly those tools. While having broader applications in fields of psychology and philosophy of sciences [2], the term *technomorphism* subsumes this creative approach regarding technological tools in composition. Briefly, it refers to the "the metaphoric use of a technological process applied in a different environment from that in which it was conceived"[3]: in the context of post-war instrumental music, it denotes, particularly, the use of compositional concepts and procedures built upon analogies with equipment, tools, and techniques introduced by electroacoustic technologies[1].

In a broader sense, in line with Gilbert Simondon's theoretical contribution [4], *technomorphism* is a process of interpretation, appropriation, and resignification of technologies embedded in modern technical objects. Just as the inventor sets down ideas and human gestures in the mechanisms and modes of operation of technical objects, *technomorphism* involves a creative reinvention of these same techniques. It relies, at first, on deciphering the mechanisms, gestures, and ideas set down in the gears and elements of the technical objects, on understanding their functionalities and characteristics. In a second moment, it involves the transposition of these features to the universe of artistic creation, where they may become compositional models, procedures, or materials[1].

With the remarkable development of computational methods for signal analysis and synthesis in recent

[1]The relationship between human imagination and the process of operation of technical objects is particularly evidenced by Simondon, in [4]: "To invent is to make ones thought function as a machine might function, neither according to causality, which is too fragmentary, nor according to finality, which is too unitary, but according to the dynamism of lived functioning, grasped because it is produced, accompanied in its genesis. The machine is a being that functions. Its mechanisms concretize a coherent dynamism that once existed in thought, which were that thought. During invention, the dynamism of thought converted itself into functioning forms. Inversely, the machine, in functioning, is subject to or produces a certain number of variations around the fundamental rhythms of its functioning, arising from its definite forms. These variations are what are significant, and they are significant with respect to the archetype of functioning, which is that of thought in the process of invention. One has to have invented or reinvented the machine if the machines variations of functioning are to become information." [p. 151]

years, it is possible to imagine compositional processes using algorithms that have as technomorphic models new approaches to computational sound synthesis/processing. In the present work, we use DSP analysis resources to implement technomorphic instrumental compositional procedures based on CSS techniques (synthesis/sampling methods that rely on computational resources for analyzing and manipulating audio signals).

3. Instrumental synthesis and computer-aided orchestration/composition

While the basic operations of serial music may be compared to the operational mechanism of computers and Turing machines, the term *technomorphism*, although applicable to the appropriation of any technologies, is usually associated with electronic means of sound synthesis. This association is presumably due to the remarkable impact that the techniques of electroacoustic music had on composers since the post-war period. The influence of these techniques is manifest in orchestral literature since at least the late 1950s, with the decisive influence of electronic processes in pieces such as *Apparitions* and *Atmosphères*, by György Ligeti. From the 1970s onwards, composers of the so-called *spectralisme*, such as Gérard Grisey and Tristan Murail, have systematically explored technomorphic processes based on techniques such as additive synthesis, feedback processes, ring modulation, and FM synthesis. The influence of studio techniques and apparatus on compositional writing of this period is described in detail in [1].

With the rapid expansion and dissemination of computational technologies several composers turned to algorithmic approaches of computer-assisted orchestration and instrumental composition. In general, these efforts have been focused on what composers such as Gérard Grisey [5, 6] and Clarence Barlow [7, 8] came to call, based on diverse aesthetic trajectories and creative elaborations, *instrumental synthesis*.

In general, such approaches consist in treating instrumental writing in resemblance with additive synthesis techniques: it consists, basically, of transcribing into the notes of the instrumental ensemble, the parameters of sinusoidal components whose specific values of frequency and amplitude can be either established through abstract processes/calculations – the case, in general, of instrumental synthesis in the works of spectralism, in the 1970s –, or from the previous analysis of audio signals using different techniques (in particular, the Fast Fourier Transform).

With the growing use of computers and digital technologies from the 1980s onwards, such researches are gradually enhanced by CAC practices. These new approaches to composition and orchestration gradually shift from very personal approaches (such as Gottfried M. Koenig's *Project 1* and *Project 2* [9]) to the employment of digital computers to more broad-use languages, libraries, and CAC environments, intended to help composers to explore algorithmic tools in their creative projects.

In the 2000s, these computational tools for music composition are already mature, being applied in creative contexts not anymore confined to the institutional borders of big European Research and Cultural Centers, like IRCAM. In this context, it is also possible to observe the remarkable development of new areas, like Music Information Retrieval, with a strong emphasis on new DSP algorithms that enable the inference of sonic characteristics. From this, there is the emergence of researches that explore the use of DSP-based processes and corpus of audio samples to open new venues for CAC and CAMO. In [10], for instance, large sets of audio files are analyzed through FFT to extract spectral peak, stored in Common Lisp lists. Target sounds are processed similarly, being subsequently compared to the database. The algorithmic process focuses on finding instrumental timbres that match, with the lowest deviation, the spectral peaks (frequency regions and amplitude) of the given sound file, allowing for the instrumental mimesis of the target audio.

Subsequent works use other DSP analysis methods to extract relevant characteristics of the instrumental corpus and the target sounds [11]. Similarly, they employ strategies to filter intermediate candidates, reducing the space search, enabling optimizations regarding computation time [12]. Finally, these works explore different algorithms and programming strategies to both perform the heuristic search and, also, to rank possible solutions. Among such approaches, it is remarkable the application of bioinspired algorithmic methods such as evolutionary algorithms, artificial immune systems, and neural networks, for instance [13, 14, 15, 16].

While the main objective of such works is, in general lines, to create tools and strategies for target-based imitative orchestration, other researches have been focusing on the creative exploration of audio descriptors in orchestral composition. In such contexts, feature-related data can be applied, for instance, as criteria to qualify, select, or manipulate instrumental sounds and combinations by taking into account characteristics that are related to high-level attributes (such as "brightness" and "roughness" for instance) [17].

4. Concatenative Sound Synthesis

While such approaches are fruitful, enabling the exploration of new creative possibilities for CAC and CAMO, we propose a third approach. Our study aims to develop algorithmic tools/strategies for computer-aided music notation based on CSS and the underlying process of juxtaposing or overlapping sound fragments selected according to their sonic features. This paradigm, already explored consciously without the direct use of computers by composers such as Helmut Lachenmann [18], consists of a different approach of exploring descriptor-based techniques to generate combinations of percussion sounds. Having established the notational resources to indicate such sounds, it becomes possible to use this process to generate percussion score excerpts.

CSS is a computational data-driven synthesis

technique with two inputs: a target sound and a collection of audios from a database. The output is a synthesized sound generated by the concatenation of audio grains[2], which are selected from a corpus according to the similarity between their features and those of the segments of given target sounds. CSS works by segmenting the input sounds into small audio "units" and analyzing them through audio descriptors and DSP algorithms. A unit selection algorithm measures the distance, in the descriptors' space, between the feature vectors obtained from the target sound and the sample corpus units [20]. The latter corresponding to the vectors with the smallest distances are sequenced and concatenated to generate the synthesized audio.

Most implementations and applications of CSS use these techniques in processes like musical mosaicing/micromontage [21, 19], sound texture synthesis [22], and real-time signal processing [23, 24]. However, there are still few researches, like [20], that apply CSS concepts and processes in contexts of generative, algorithmic, or computer-aided composition – particularly those focused on instrumental music.

5. Technomorfic Implementation of Concatenative Sound Synthesis

The application of CSS as a technomorphic model for instrumental composition is based on the correlation between relevant features of audio frames of a target sound and those of samples available in a database/corpus. Instead of using the corpus samples to synthesize sounds as in usual CSS processes, the technomorphic adaptation of this synthesis technique uses the respective symbolic information of corpus sounds to inform or generate musical notation. The resulting notation prescribes instrumental indications and mixtures whose resulting sounds are expected to have a degree of similarity to the spectral and temporal structure of the target audio.

In our study, we structured the architecture and the main stages of the technomorphic adaptation of CSS. Our implementation of the process uses the Python programming language due to the diversity of available libraries, packages, and modules. Among the libraries that we are already exploring for our purpose or that may be helpful to further developments of our research are those dedicated to audio analysis (librosa, pyAudioAnalysis, Essentia), musical score notation/manipulation (abjad, music21), and DSP/synthesis (audiolazy, pyo, supriya). Beyond those, popular packages dedicated to machine-learning and linear algebra processes available in Python are promising for descriptor-based processes in CAC and CAMO contexts.

Currently, our implementation comprises four steps:

1. segmentation of both the target sound and of the database's audio samples into *attack* and *decay* segments;
2. analysis of the segments by a set of audio descriptors;
3. selection/ordering of corpus samples segments according to the degree of similarity between them and those of the target sound;
4. music notation/transcription according to the selected corpus samples.

The following subsections describe in more detail this preliminary approach to CSS as a technomorphic model. In our tests, the target audio is an electroacoustic soundscape consisting of different percussive/impulsive events. The samples corpus is composed of simple percussion instrument samples (conga, bongo, tom-tom, snare, cowbell, shakers and cymbal).

5.1. Segmentation

Initially, every audio is pre-processed by a method of Harmonic Percussive Source Separation[3] [25, 26], which can accentuate transient and attenuate harmonic sounds. Further, a segmentation algorithm computes the root mean square (RMS) envelope of each audio file and applies an onset detection function[4] and a peak-tracking function[5]. The purpose is to identify individual *sound objects*. Since we have so far worked with relatively simple percussive/impulsive sounds, it is considered that each *sound object* comprises two main parts: the *attack* and the *decay* [27].

Supposing that there are meaningful spectral variations between these two parts that could enrich the diversity of the data, the algorithm splits each sound object into an *attack* and a *decay* fragment according to the location of the onset, peak, and offset frames. The *attack* segment corresponds to the frames between the onset and the peak of the RMS envelope. The *decay* segment spans from the peak frame until an offset frame with an RMS value inferior to an arbitrary threshold[6] (Figure 1). However, since the target sound can comprehend sustained events or sequences of overlapping gestures, the RMS envelope of the segmented sound objects may not always present a value below the determined offset threshold. In these cases, the *decay* section will span until the frame right before the onset of the following event (Figure 2).

5.2. Analysis

After splitting every *sound object* into an *attack* and a *decay* segment, an algorithm built upon a determined set of audio descriptors analyzes frames of the samples to extract feature-related data as vectors. In this specific test, the audio analysis produced for each frame a vector containing 13 MFCC and 1 RMS coefficient. If an audio fragment

[2] A more detailed explanation of the mechanism of CSS and the differences between it and granular synthesis can be found at [19].

[3] For this, we used the function `effects.hpss()`, from Librosa.

[4] Namely, `onset.onset_backtrack()`, from Librosa.

[5] Namely, `util.peak_pick()`, from Librosa.

[6] The offset is computed considering RMS envelopes of the original audios since the pre-processed ones would have their reverberation considerably attenuated.

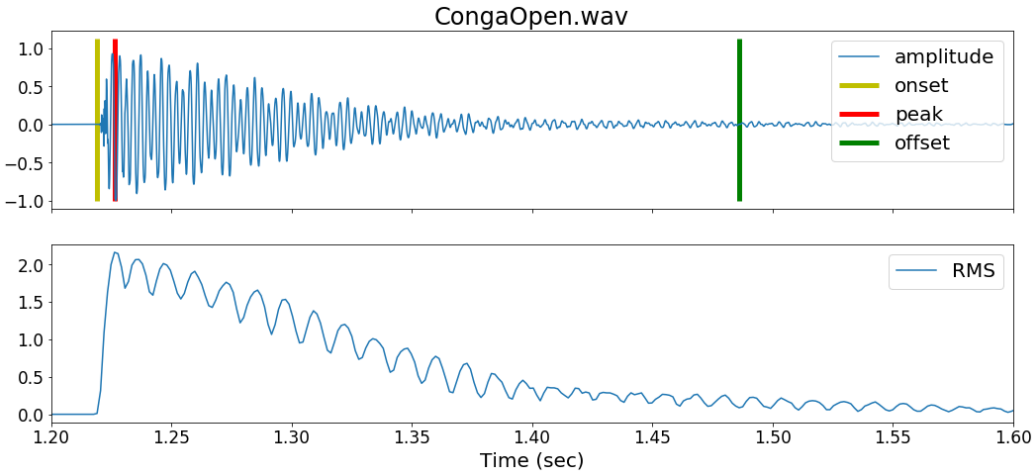

Figure 1: Segmentation of a one-hit conga sample.

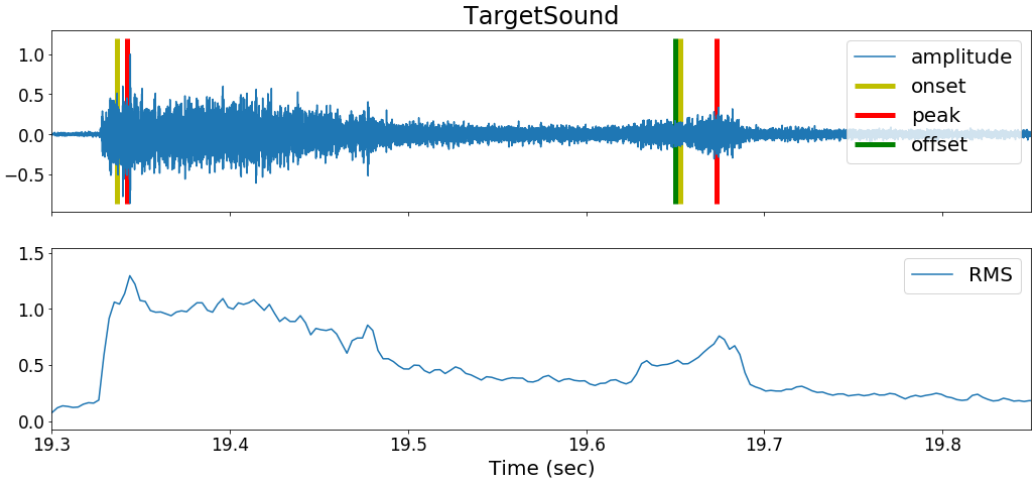

Figure 2: Segmentation of an extract of the target sound.

comprises two or more frames, a function averages corresponding coefficients into one vector.

Formally, let Ω_c and Ω_T be the abstract space of the corpus samples and target sound objects after their segmentation into *attack* and *decay* segments. Then, an element β in those spaces would be understood as a pair (β_1, i) where β_1 represents the sound object and $i \in \{0, 1\}$, where 0 represents the *attack* component of the sound and 1 the *decay*.

We first transform these spaces into a feature space. The vectorization is done by summarizing audio information using audio descriptors, where each of these descriptors will be treated as a dimension. In other words[7],

$$\begin{aligned} f &: \Omega_T \longrightarrow [0,1]^d \times \{0,1\}, \\ y_i &\triangleq f(\beta) = (\bar{y}_1, \bar{y}_2, \cdots, \bar{y}_d, i) \end{aligned} \tag{1}$$

where $i \in \{0, 1\}$, d is the number of descriptors, and the entries $\bar{y}_j$ are the average of the j-th descriptor of the vectors extracted from each frame of β.

[7] Since we are using the same descriptors for Ω_c, the transform for this space follows analogously.

Notice that the vectorial space is normalized. Since we will use distances to approximate feature vectors, we need to rescale the space to avoid bias towards certain dimensions.

5.3. Selection

Our objective is to approximate elements of Ω_T using Ω_c. In order to convey a notion of distance between the spaces, we chose an Euclidean based approach.

Let $y_i \in f(\Omega_T)$ be a sound object which we wish to approximate using elements of $f(\Omega_c)$. Then such minimizer could be defined as

$$x_i^* \triangleq \underset{x_j \in f(\Omega_c), j=i}{\arg\min} D(y_i, x_j) \tag{2}$$

where D is some distance function.

Since Ω_c is a finite space, we may simply use an algorithm to exhaustively search Ω_c, calculate the distances between the elements and y_i and choose the minimizer. Notice that the optimization is restricted to sounds of same kind (attack or decay).

The percussion fragments related to the vectors with the smallest distances are then selected and sequenced according to the respective target sound segments (Table 1). The information related to the origin and kind of the segments is embedded in their names, which follow the given structure:

`<sloc>_<fname>-<kind>_<sdur>`

where `<sloc>` is the fragment location, in samples, in the audio file from which it was extracted, `<fname>` is the name of the audio file, `<kind>` labels the fragment as attack (ATT) or decay (DEC), and `<sdur>` is the segment duration, in samples.

Target Sound Fragment	Distance	Selected Percussion Fragment
22660_TS-ATT_3914	0.169	23690_Cowbell1-ATT_2266
26574_TS-DEC_12772	0.133	44496_SnareBright-DEC_8240
39552_TS-ATT_1442	0.792	23896_SnareBright-ATT_412
40994_TS-DEC_7004	0.076	24576_CymbalStrike2-DEC_219136
...	...	...

Table 1: Attack and decay percussion segments sequenced according to target sound fragments. Distances are normalized between 0 and 1.

To check for acoustic similarities between the target sound and the instrumental sequence, the latter is concatenated into an audio file, as would occur in a traditional process of CSS. This procedure allows the composer to grasp, by hearing, what types of sonorities were selected according to the parameters passed to the segmentation and analysis algorithms. Thus, the composer may experiment with different parameters, such as distinct sets of audio descriptors, and then listen to the acoustic differences between the segment sequences produced in each run before choosing one or more of them to apply in a compositional process. This approach seems interesting to us as it allows flexible applications of the computational tool in compositional contexts. Such an approach enables both generative/algorithmic applications and more dialogical explorations of the computational processes, standing precisely between the extremes of "manual composition" and the "composing machine" that, as Laske points out, situate CAC [9].

5.4. Notation

After these stages, the final step consists of determining the appropriate symbols and resources to notate the percussion sounds in a musical score. This is done using Python and LilyPond data structures and functions to create the music score structure, notational strategies (including rhythm quantization and instrumental technique indications, for instance), and the association between corpus sounds and graphic symbols.

This final step uses the flexibility of Abjad [28, 29], a Python package that enables the algorithmic manipulation of symbolic music structures that are subsequently rendered using LilyPond [30]. While requiring the further formalization of compositional decisions, this approach allows for a highly customizable and flexible algorithmic notation. Figure 3 demonstrates an example of a raw percussion score generated through this process.

6. Conclusion

We have documented a preliminary approach to CSS as a technomorfic model for music composition. Due to the notational purpose, instead of segmenting input audios into units of fractions of a second, as commonly occurs in CSS, a segmentation algorithm splits each audio into *attack* and *decay* fragments. The segments of a target sound are analyzed and compared to those of percussion samples (corpus), allowing for the sequential selection of sound combinations and the respective data related to their duration, instrumentation, and temporal onset/offset. These parameters enable the creative use of the generated data in compositional contexts, enabling CAC approaches to the technomorphic adaptation of CSS.

By using open-source tools and high-level programming and analysis resources, the purpose of our study is to explore descriptor-based CAC/CAMO processes that allow composers and researchers to manipulate particularities of the diverse algorithmic tools involved. Thus, while our approach does not intend to offer a ready-to-use tool for target-based orchestration, it proposes an exploratory integration of different computer tools in the context of feature-based computer-aided composition. Besides a more conscious manipulation of the computational tools applied in CAC/CAMO contexts, this strategy seems to have the potential to integrate research and artistic practices and to enable creative efforts in using other computational tools, such as machine-learning packages/libraries for example.

Regarding the presented implementation, we plan to keep a creative and flexible approach to this model. Further developments comprise other strategies of segmentation for the processing of instrumental samples other than percussion, the structuration of larger sample corpus (using specialized database formats), different approaches of instrument superposition (allowing for the creation of instrumental textures to create "sound types" and orchestrations), and the exploration of machine-learning and other heuristic processes.

References

[1] Tatiana Catanzaro. *Transformações na linguagem musical contemporânea instrumental e vocal sob a influência da música eletroacústica entre as décadas de 1950-70*. 7 Letras, Rio de Janeiro, RJ, 2018.

[2] Bryan Holmes. *Tecnomorfismo em música: uma visão teórica e prática*. PhD thesis, Unirio, Rio de Janeiro, 2019.

[3] Peter Niklas Wilson. Vers une 'écologie des sons': Partiels de Gérard Grisey et l'esthétique du groupe de l'Itinéraire. *Entretemps*, 8:56–81, 1989.

[4] Gilbert Simondon. *On the mode of existence of technical objects*. Univocal Pub, Minneapolis, MN, 2017.

[5] Gérard Grisey. À propos de la synthèse instrumentale. In Guy Lelong, editor, *Écrits ou L'invention de la musique spectrale*. MF, Paris, 2008.

[6] Gérard Grisey. Structuration des timbres dans la musique instrumentale. In Guy Lelong, editor, *Écrits ou L'invention de la musique spectrale*, pages 89–120. MF, Paris, 2008.

[7] C. Barlow. On the spectral analysis of speech for subsequent resynthesis by acoustic instruments. *undefined*, 1998.

[8] Clarence Barlow. Music Derived from Other Sources. *Journal of Science and Technology of the Arts*, 13(1):21–44, April 2021. Number: 1.

[9] Otto Laske. Composition Theory in Koenig's Project One and Project Two. *Computer Music Journal*, 5(4):54–65, 1981. Publisher: The MIT Press.

[10] David Psenicka. SPORCH: An Algorithm for Orchestration Based on Spectral Analyses of Recorded Sounds. In *Proceedings of the 29th International Computer Music Conference*, pages 1–4, Singapore, 2003. Michigan Publishing.

[11] Grégoire Carpentier, Damien Tardieu, Gérard Assayag, Xavier Rodet, and Emmanuel Saint-James. Imitative and Generative Orchestrations Using Pre-analyzed Sound Databases. In *Proceedings of the Sound and Music Computing Conference*, pages 1–8, Marseille, 2006.

[12] Grégoire Carpentier and Jean Bresson. Interacting with Symbol, Sound, and Feature Spaces in Orchidée, a Computer-Aided Orchestration Environment. *Computer Music Journal*, 34(1):10–27, March 2010.

[13] José Abreu, Marcelo Caetano, and Rui Penha. Computer-Aided Musical Orchestration Using an Artificial Immune System. In *Proceedings of the 5th International Conference on Evolutionary and Biologically Inspired Music, Sound, Art and Design - Volume 9596*, pages 1–16, Berlin, Heidelberg, March 2016. Springer-Verlag.

[14] Jon Gillick, Carmine-Emanuele Cella, and David Bamman. Estimating Unobserved Audio Features for Target-Based Orchestration. In *ISMIR*, 2019.

[15] Carmine Emanuele Cella, Daniele Ghisi, Vincent Lostanlen, Fabien Lévy, Joshua Fineberg, and Yan Maresz. OrchideaSOL: a dataset of extended instrumental techniques for computer-aided orchestration. *arXiv:2007.00763 [cs, eess]*, July 2020. arXiv: 2007.00763.

[16] Marcelo Caetano and Carmine E. Cella. Imitative Computer-Aided Musical Orchestration with Biologically Inspired Algorithms. In Eduardo Reck Miranda, editor, *Handbook of Artificial Intelligence for Music: Foundations, Advanced Approaches, and Developments for Creativity*, pages 585–615. Springer International Publishing, Cham, 2021.

[17] Ivan Eiji Yamauchi Simurra. *Contribuição ao problema da orquestração assistida por computador com suporte de descritores de áudio*. Doctor of Music Thesis, Universidade Estadual de Campinas, Campinas, 2016.

[18] Helmut Lachenmann. Klangtypen der Neuen Musik. In *Musik als existentielle Erfahrung: Schriften 1966-1995*, pages 1–20. Breitkopf & Härtel, Wiesbaden, 1996.

[19] Bob L. Sturm. Adaptive Concatenative Sound Synthesis and Its Application to Micromontage Composition. *Computer Music Journal*, 30(4):46–66, December 2006.

[20] Gilberto Bernardes de Almeida. *Composing Music by Selection: Content-Based Algorithmic-Assisted Audio Composition*. Doctor of Philosophy, University of Porto, Porto, 2014.

[21] Aymeric Zils and François Pachet. Musical Mosaicing. *Proceedings of the COST G-6 Conference on Digital Audio Effects*, pages 1–6, 2001.

[22] Diemo Schwarz and Axel Roebel. Concatenative sound texture synthesis methods and evaluation. In *Proceedings of the 19th International Conference on Digital Audio Effects (DAFx-16)*, Brno, Czech Republic, 2016.

[23] William Brent. *A timbre analysis and classification toolkit for pure data*. Ann Arbor, MI: MPublishing, University of Michigan Library, 2010.

[24] Diemo Schwarz, Gregory Beller, Bruno Verbrugghe, and Sam Britton. Real-Time Corpus-Based Concatenative Synthesis With CATART. In *Proceedings of the 9th Int. Conference on Digital Audio Effects*, pages 1–7, Montreal, Canada, 2006.

[25] Derry Fitzgerald. Harmonic/Percussive Separation Using Median Filtering. In *Proceedings of the 13th International Conference on Digital Audio Effects*, pages 1–4, Graz, Austria, 2010.

[26] Jonathan Driedger, Meinard Müller, and Sascha Disch. Extending Harmonic-Percussive Separation of Audio Signals. In *Proceedings of the 15th International Conference on Music Information Retrieval (ISMIR)*, pages 611–616, Taipei, Taiwan, 2014.

[27] Juan Pablo Bello, Laurent Daudet, Samer Abdallah, Chris Duxbury, Mike Davies, and Mark B Sandler. A Tutorial on Onset Detection in Music Signals. *IEEE Transactions on Speech and Audio Processing*, 13(5):1035–1047, 2005.

[28] Trevor Baca, Josiah Wolf Oberholtzer, J. Trevino, and V. Adán. Abjad: An open-source software system for formalized score control. In *Proceedings of The First International Conference on Technologies for Music Notation and Representation*, 2015.

[29] Gregory Evans. *An Introduction to Modeling Composition Through Abjad's Model of Music Notation*. Master's thesis, University of Miami, Miami, 2019.

[30] Han-Wen Nienhuys and Jan Nieuwenhuizen. LilyPond, a System for Automated Music Engraving. In *Proceedings of the XIV Colloquium on Musical Informatics (XIV CIM 2003)*, volume 1, pages 167–171, Florence, 2003. Citeseer.

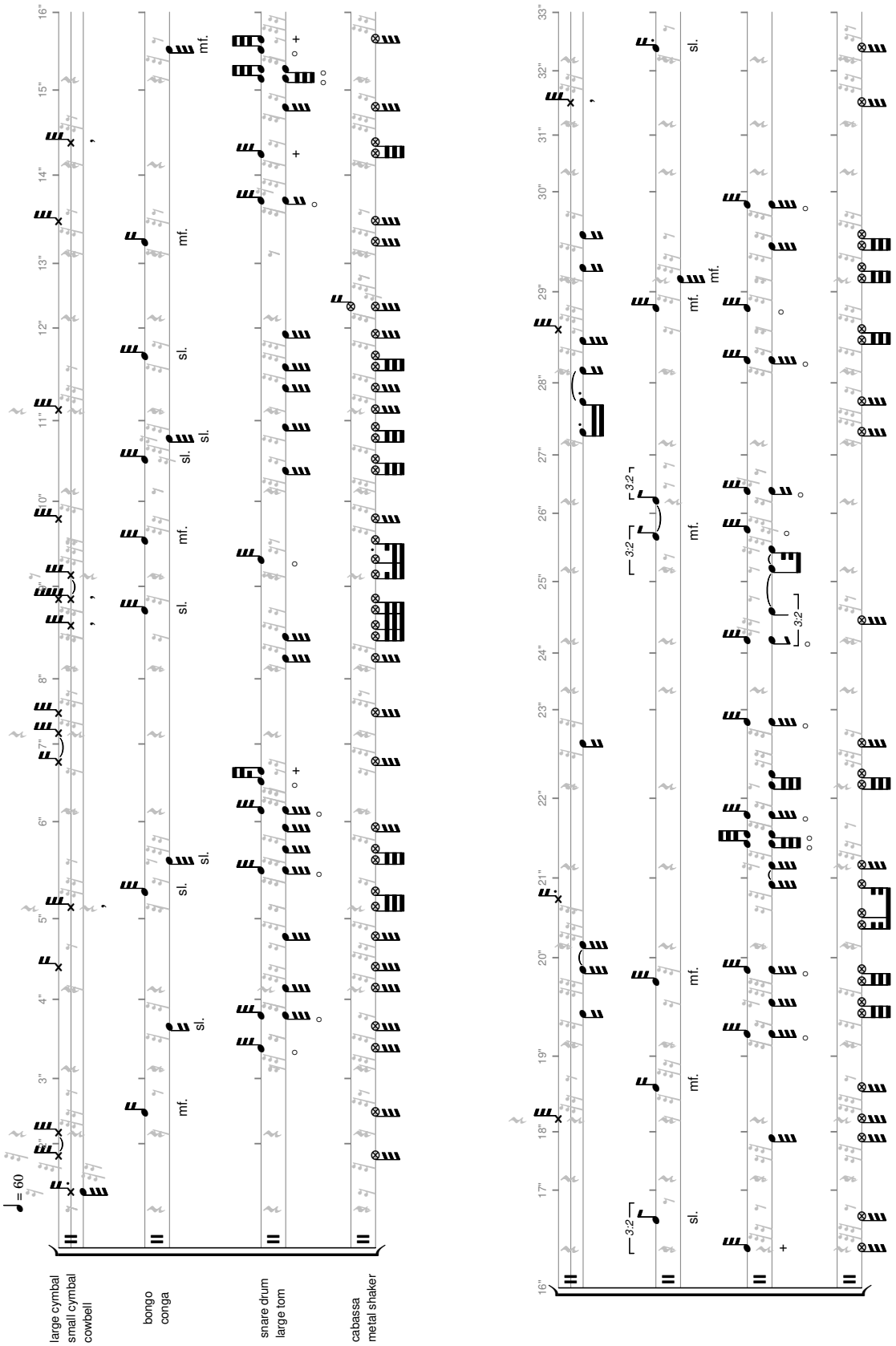

Figure 3: Abjad/LilyPond sketch with the raw orchestration of the target sound using the technomorphic model of computer-aided composition.