# Real-Time Composition, why it still matters: A look at recent developments and potentially new and interesting applications

**Carlos Guedes**
New York University Abu Dhabi
carlos.guedes@nyu.edu

## ABSTRACT

*In this paper I present a definition of Real-Time Composition (RTC) as well as framework for classifying systems that enable this type of compositional approach. I also present four examples of RTC systems in which I was involved with their development, and discuss why is it important to look at RTC as a framework that can provide new interesting and potentially revolutionary approaches to musical education and enculturation.*

## 1. INTRODUCTION

I define real-time composition (RTC) as a "*Compositional practice* utilizing *interactive* music systems in which *generative* algorithms with a *non-deterministic* behavior are manipulated by a user during performance" [1]. The terms in italic also define important keywords about RTC: it is a performative practice, it is interactive, it is generative, and non-deterministic. This resonates with original ideas of Joel Chadabe [2] and with more recent thoughts by Arne Eigenfeldt [3][4] about RTC as constituting a performance ecosystem that is different from improvisation with electroacoustic instruments. Chadabe [2] defines interactive composition as a process involving performable interactive computer systems in performing and composing music. He considers interactive composition as a two-stage process consisting of (1) creating an interactive composing system, and (2) simultaneously performing and composing with the system as it functions. Eigenfeldt [4] defines real-time composition as "the application of musical agents to interact in musical ways, during performance" (p. 146). He articulates in a rather detailed way the differences between real-time composition and (computer) improvisation, and is interested in situating this practice in the realm of live electroacoustic music performance.

A more recent look into emerging software applications that engage with RTC as defined above, including current applications of generative music in education, for example as discussed in [5], makes us realize that RTC seems to be more widespread, extending its applications beyond "art" music and opening up new and engaging opportunities for music making by lay people, and for music education. The increase in recent years in interest on musical metacreation, either through specialized publications on the subject (e.g. [6]), or through the creation of specialized symposia on the subject (e.g. the Workshop on Musical Metacreation) has bolstered critical reflection over the general topic of music AI and user-interaction with automated/generative music systems. A more general look at RTC as an emerging practice across many domains where computer-generated music is utilized thus seems to be an appropriate thing to do.

### 1.1 Emerging practices

The myriad of situations in which one can find examples of real-time composition as defined above provides the main motivation for this paper: (1) the emergence of software applications for smart phones or portable game consoles employing generative music algorithms whose behavior is controllable by users; (2) the appearance of sequencing software that allows non-linear sequencing and its control in real time (e.g. Ableton Live); as well as (3) generative music modules in commercial sequencing software that allows the control of music by specifying certain high-level parameters (e.g. Logic's Drummer) denote pertinent changes in the practice of computer-generated electronic music. If we add to this the current resurgence of modular, voltage-controlled instruments in what it seems to be a return of the live electronic music from the 1960s now enriched by the digital revolution, one realizes that something different is in the air.

These changes have to do in a great part with a progressive shift from using the computer as a machine that can provide sonic results otherwise unachievable by other means, towards the increasing use of the computer as some sort of musical companion with a musical behavior of its own in interaction with its users. Moreover, twenty years ago, the use of the computer as an interacting entity in musical performance could only be appreciated in specialized computer music concerts in certain (restricted) environments like universities or conferences. Nowadays it is not uncommon to carry an application on your smartphone that produces music interactively through tapping or swiping gestures on the phone's touchscreen.

The appearance of applications containing generative music algorithms whose behavior can change in real time while responding musically to some user input has become increasingly common. Games, interactive installations, sequencing/composition software are some of the examples. The time is thus ripe to (1) try to better understand this emerging compositional-performative practice that cuts across different domains and genres, which relies on real-time interaction with algorithms that produce a musical output; and (2) to start providing a theoretical framework for addressing this new practice.

Instead of discussing yet another ontological framework for RTC I will focus essentially in providing a taxonomy for classifying RTC systems and suggest a generic approach for RTC systems design. Eigenfeldt [3][4] provides compelling arguments for situating this practice in the realm of composition. I think that the ethos of real-time composition relates to algorithmic composition, to the creation of a metalevel of operation in RTC systems, and to the sonic spaces that can be navigated during a performance in this situation.

Essentially, an RTC system can be seen as a peculiar combination between algorithmic composition, interactive music systems, and digital musical instrument design. Interactive music systems provide the possibility of modifying the behavior of an algorithm in real time, and can enable a metalevel approach to composition through the possibility of interactive/real-time control of generative algorithms [7][8]. Digital musical instruments enable the creation of complex mediation spaces between physical gesture and sonic result in which these algorithmic approaches can be utilized [9].

### 1.2 Why is it composition?

The first question to pose is why should real-time composition be considered composition and not improvisation or something else. The term *interactive composition* as coined by Chabade [2] is definitely an alternative designation and one could replace the term *real-time* with *interactive* if instead one wants to emphasize the interactivity. I personally prefer to use *real-time*. But why call it *composition?* Aside from the pertinent arguments advanced by Eigenfeldt [3][4], there are still other things to consider.

The notion of what is a musical instrument and what types of class of instrument are there has been dramatically challenged with the advent of electronic and computer-based instruments (cf. [10]). On one side, there is the question if a musical instrument can (still) be considered a single sound-producing body as nowadays — especially with the use of distributed sensing, processing, computational and streaming technologies — this may not be always the case. On the other side, the emergence of what Chadabe [9] has termed *interactive instruments,* or indeterministic electronic instruments, blurs the notion of what musical instrument performance *is* in the traditional sense. In interactive instruments, the mediation is done through an interactive music system. As Drummond [11] rightly and succinctly puts it "[i]nteractive systems blur these traditional distinctions between composing, instrument building, systems design and performance." (p. 124). The complexities of relations that can be established with interactive music systems challenge the traditional paradigms in music performance, composition, and instrument design. Brown, Eldridge and McCormack [12] criticize the *acoustic paradigm* often used as a metaphor to describe the types of relationship that can be established between the users and these systems (see for example [7, 13]), and suggest new paradigms for addressing the new interconnections that can be established between software systems in these new situations.

The fact is that interactive music systems contribute to blur these distinctions — which are often imperceptible when one watches a performance. The performers of a certain system may even not grasp what the system is doing while they're performing it — such as in the case of certain games or applications. This makes it hard to really understand where does one establish the boundary between digital instruments that simulate traditional instruments and RTC systems, or other interactive instruments that are *not* RTC systems. Moreover, if whatever one is doing while interacting with the system should be considered performing, improvising, or composing.

There is however an important characteristic that stands out in this type of practice that frame it within the realm of musical composition in my view, which is the use of algorithms that provide a musical space that can be explored interactively. Dodgde and Jerse [14] acknowledged two broad categories in which algorithmic composition with computers fall into: stochastic music, in which events are generated based on some statistical representation; and music in which the computer is used to calculate permutations of predetermined conditions. In both situations the computer is providing musical, navigable spaces that bear the characteristics defined by and implemented in the algorithm.

This navigable space is typical of algorithmic composition with computers, and was identified by Xenakis [15] on his famous account of his first experience with the computer. The control/alteration of parameters in algorithmic computer music provides the possibilities for navigation of the musical space, whose limits are defined by the ranges of values in the parameters.

Heinrich Taube [16] considers the metalevel as a representation of *the composition of the composition* in algorithmic music: "A metalevel representation of music is concerned with representing the activity, or *process*, of musical composition as opposed to its artifact or score" (p. 3). He makes a pertinent distinction between computer-assisted, automatic, and computer-based composition as three different possible ways to engage with algorithmic composition. In short, he considers computer-assisted composition a situation where the computer facilitates compositional tasks such as computing pre-compositional data, and as a simulation tool; automatic composition relates to systems that compose music independently (e.g. David Cope's EMI). Finally, computer-based composition,

> [M]eans to use the computer to explicitly represent compositional ideas at a level higher than the performance score. An explicit metalevel representation means that the relationships and processes that constitute a composition (the composition of the composition) are represented inside the machine apart from the composer thinking about them. (p. 5)

Taube does not consider the temporal scale at which computer-based music can occur. Although he may not even be considering the real-time application of these concepts the properties of computer-based music he mentions can certainly be found in RTC systems.

With this short theoretical discussion as a background, I will now define RTC systems provide a taxonomy for their classification and describe their anatomy.

# 2. RTC SYSTEMS: DEFINITION, TAXONOMY AND ANATOMY

Real-time composition systems are interactive music systems that enable composition in real time. These systems exist in the form of standalone applications, plug-ins, or even as libraries or programming environments that can facilitate the creation of such systems. They can operate at the sub-symbolic or symbolic levels. The majority of existing RTC systems operate at the symbolic level.

Two levels of utilization of these systems can be identified: systems designed for common/lay users and systems for specialists. Systems for common/lay users are easy and simple to operate and the processes inherent to music/sound generation are hidden from the users — e.g. *Bloom* [17]. Systems for specialists require specific knowledge for their operation. These can be programming environments, specialized libraries or even commercial software designed to be operated by users who have acquired specific knowledge for their operation — e.g. Karlheinz Essl's RTC Lib,[1] Max,[2] or Supercollider.[3] Developments in user interface design and availability sometimes blur the differences between the two levels. For example, the Drummer plug-in exists in both Apple's GarageBand and Logic Pro X, which are software sequencers dedicated to two different types of user representing the two ends of this spectrum. However, what remains typical of systems designed for specialist use is the fact that these systems are more malleable and reconfigurable.

A RTC system is essentially a type of conceptual approach for the design of interactive music systems with a generative response. The fact the response should be non-deterministic is essential to promote interaction between the users and the systems. Another important aspect is the definition of parameters that can affect the musical generation over time. A parametric approach to system design is essential in real-time composition systems and an important feature in non real-time algorithmic composition (Cf. [18]). The manipulation over time of parameters controlling the generation is what makes real-time composition possible as it enables on-the-fly interventions over what is being generated by the system.

A RTC system should possess at least two components: a musical space that is defined by a generative algorithm, which provides the musical material that can be obtained and transformed by navigating that space, and parameter controls that provide access to that space or its features.

A clear and simple example of a RTC system as defined above is Logic Pro' s Drummer plug-in (figure 1). Each Drummer is characterized by a drumming style and personality. The generation of drumming patterns can be controlled by the user in real time within the style features performed by the drummer. A bi-dimensional plane in which the horizontal axis spans from simple to complex and the vertical plane spans from soft to loud provide a *musical space* that can be traveled by the users during the algorithm's execution.

[1] http://www.essl.at/works/rtc.html
[2] http://www.cycling74.com
[3] http://supercollider.github.io

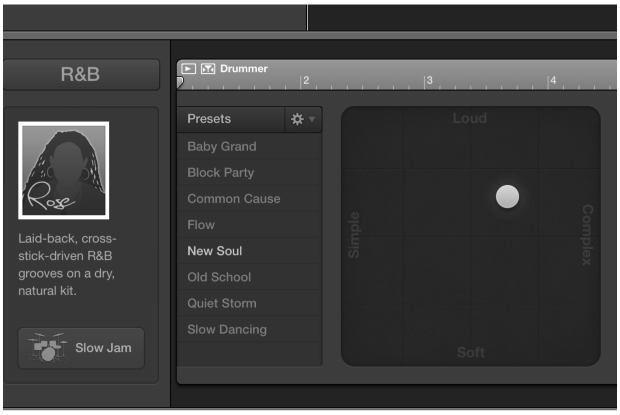


**Figure 1**. Logic Pro's Drummer "Rose," which performs "[L]aid-back, cross-stick-driven R&B grooves on dry, natural kit." Notice the control parameters "soft-loud" (y-axis) and "simple- complex" (x-axis) that can be used in real time to alter the behavior of the algorithm.

In the upcoming section I will describe some of the work I have been involved with in the creation of RTC systems both for lay and specialized users, that follows the framework described above.

# 3. EXAMPLES OF THE APPLICATION OF THE PROPOSED FRAMEWORK

I stumbled into real-time composition in my doctoral work [19], where I developed software to promote real-time computer-mediated collaboration between dancers and musicians [20, 21, 22]. This work consisted in the creation of a system that extracted "musical cues" from dance movement — rhythms in dance that possess qualities of musical rhythms — and could be used to drive several generative algorithms with a stochastic behavior, which in turn were controlled by the musician. This was a personal breakthrough for me as a composer and since then I have been researching computational systems that can facilitate real-time composition to different types of user. The goal of this research is two-fold: 1) the creation of automatic music generation and interactive music systems to facilitate musical creation and expression by non-specialists; 2) the possibility of creating compositional musical spaces that can be explored interactively and impossible, or hard achieve, by other means.

Below I present three examples of research I was involved with, which follow the framework for RTC system design I introduced above.

### 3.1 The rhythmicator and recombinator applications

kin.rhythmicator [23, 24] and kin.recombinator [25] are part of the Kinetic toolbox set of externals for Max and Pd.[4] The kin.rhythmicator is a stochastic rhythm generator in which the user can specify the meter and the metrical subdivision of the rhythm to be generated by the application. The rhythm generation is not bound by any specific style and it uses at its core Clarence Barlow' s metric indispensability algorithm [26]. It is up to the user

[4] http://smc.inesctec.pt/kinetic/?page_id=19

to modify and control the output of the algorithm during a performance by altering descriptive musical parameters that produce perceivable changes in the output such as the density of events per bar, the amount of syncopation, the degree of metrical strength, the amount of variation in generation, and of course the meter itself. In this sense, the algorithm behaves like a musical companion that responds musically to requests by the user made in musical terms. One important feature of the rhythmicator application is the complexity map, a graphic interface that gives the user the possibility to continuously modify the rhythm generation different degrees of variation and different degrees of phenomenal accent that go from syncopated to accents that destroy the feeling of metrical regularity.

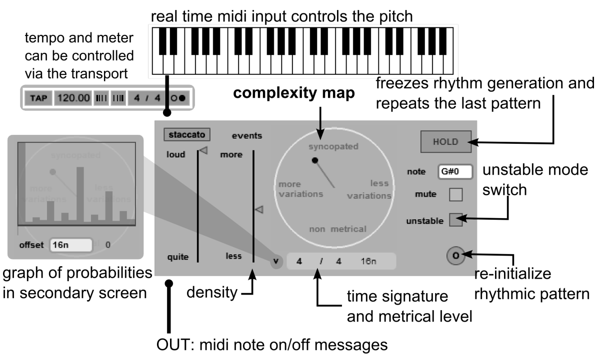


**Figure 2**. The user interface for the kin.rhythmicator application.

The kin.recombinator application [25] generates rhythmic patterns by recombining existing ones. The recombination process consists of playing back MIDI drum loop files by selecting portions of these files at regular intervals. An analysis of the files is performed prior to the recombination, in order to sort them according to their complexity and, in this way, better control the resulting rhythms. The user interface for the kin.recombinator is rather simple and consists of a range slider (Max's rslider object) that goes from "simple" to "complex." At any position the user can widen the slider to a given range and play portions from adjacent loops.

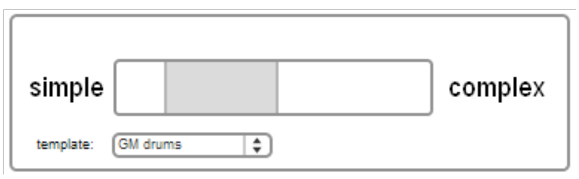


**Figure 3.** User interface for the kin.recombinator application

### 3.2 GimmeDaBlues

GimmeDaBlues [27] is an application for iOS devices that allows anyone to play jazz keyboard and solo instruments along a predefined harmonic progression using the multi-touch properties of the iOS devices. While the user plays keyboard and/or solo instruments, the application automatically generates the bass and drums parts, responding to the user's activity. The application provides a musical space that evolves over time according to the harmonic progression in blues style. The dynamic mapping of the notes and chords available in the graphical interface provides an intuitive and natural way to play otherwise complex chords and scales, while maintaining a physical playability that will be familiar to experienced keyboard players.

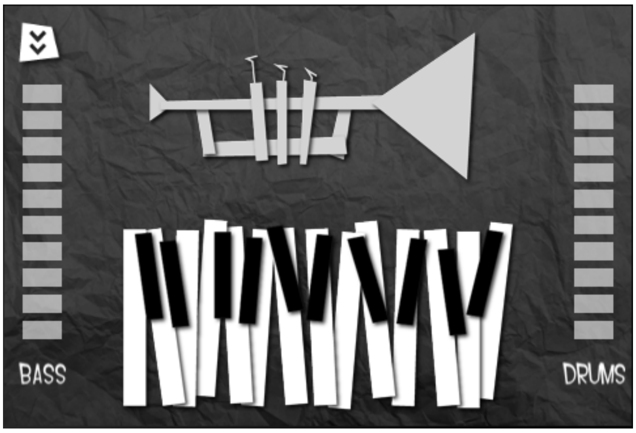


**Figure 4.** User interface for GimmeDaBlues. Users can play the piano by tapping on the piano keys creating voicings with different density (number of notes) and dynamics depending on where do they tap. If they tap on the trumpet portion of the GUI they can generate melodies that are in the scales corresponding to the harmony that is played.

### 3.3 CaMel

CaMel is the latest project I have been involved with for the creation of a real-time composition system that follows the framework proposed above. It is a generator of rhythmic sequences in Carnatic music style, currently in the form of a Max patch. The program is able to generate different sequences in adi tala, and the user can navigate a space that contains clusters of patterns that are grouped by similarity. CaMel is a data-driven generative model, whose patterns were extracted from more than 6 hours of recordings of Carnatic music percussion performed on the mridangam and on the kanjira. These recordings were annotated using Sonic Visualiser, the strokes were encoded into register, duration (IOI), and dynamics, and then parsed into patterns using Godfried Toussaint's Mutual Nearest Neighbor grouping algorithm [28].

We used the bags of words approach, which is commonly used in document classification and clustering as a tool of feature generation. We transformed the text of the groupings to a "bag of words" vector representing the frequency of occurrence of each unigram term in the groupings. This leaded to the generation of a feature matrix for all the groupings, which further used for clustering analysis in terms of similarity using the K-means clustering approach. The clusters were then mapped in 2D space using t-SNE [29].

The user interface depicts several dots in different shades of grey. Each dot represents a cluster of patterns, and darker dots represent clusters containing more patterns (Figure 5).

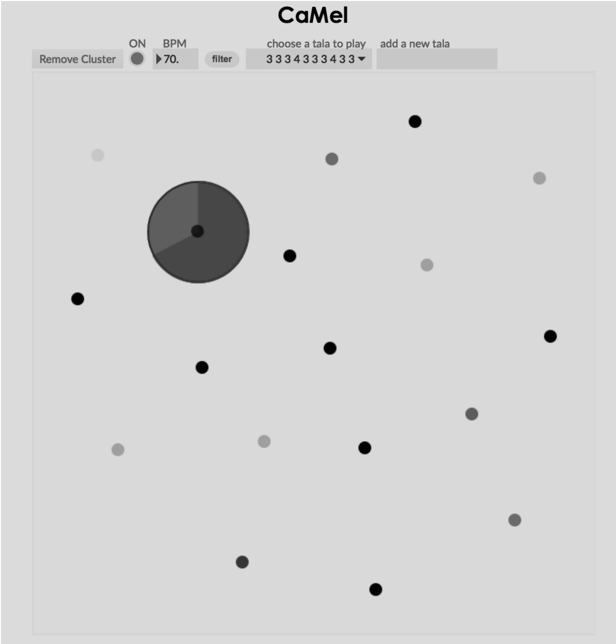


**Figure 5.** User interface for CaMel.

When a cluster is selected, the program performs elements of that cluster with a degree of variation that is represented by the radius of the circle around the dot.

a)  b)  c) 

**Figure 6**. Different degrees of variation in the performance of elements from the cluster. 6a) represents no variation, 6b), some variation, and 6c) maximum variation.

During a performance with CaMel the user can navigate this space of rhythms by selecting a cluster to be performed as well as the degree of variation. Moving within clusters that are nearby creates smooth variations between the generated rhythms. Moving to more distant clusters will create more audible differences on the rhythms being generated.

# 4. CONCLUSION: POTENTIALLY NEW AND INTERESTING APPLICATIONS

In this paper I presented a definition for Real-Time Composition (RTC) followed by a brief discussion about emerging practices that expand this activity beyond of what has been discussed in [2, 3, 4] who have described this activity within electroacoustic art music performance. This definition of RTC frames this practice in the realm of composition with interactive music systems that have a generative and non-deterministic response. This type of response is crucial in order to promote interaction with these environments. RTC systems are defined as a subset of interactive music systems that contain at least two components: (1) a dynamic musical space that gets defined and shaped by a generative algorithm, and (2) control parameters that provide access to that space during the execution of the algorithms in real time. This framework seems to be present in many recently available applications, and the use of RTC seems to correspond to the increasing use of computers as musical companions in music making over the past 20 years. The use of RTC seems to have extended beyond the realm of electroacoustic art music, and two levels on the design of such systems can be distinguished: systems designed for lay users and systems designed for specialized users.

I presented four examples of systems whose development I have been involved with: the rhythmicator and recombinator applications, GimmeDaBlues, and CaMel. Of these, GimmeDaBlues is a system designed for lay users, and CaMel is a system that can be used by lay users for experiencing Carnatic-style percussive sequences. The rhythmicator and recombinator applications, and CaMel, are oriented for specialized users in order to explore novel rhythmic spaces interactively and achieve musical results otherwise difficult or impossible to obtain.

This work on the design of RTC systems for lay and specialized users can be inscribed in a recent trend of work involving interactive music systems that includes applications such as *Electroplankton* [30], *Bloom* [17], *ixiQuarks*,[5] or the Drummer plugins developed by Apple for the Logic Pro X and Garageband applications.

In this new world of RTC, I find of particular interest the possibilities opened up by the systems designed for lay users. Chadabe [9] recognized the potential of interactive instruments to define new ways for the public to experience music. Applications such as *GimmeDaBlues* and *CaMel* were designed having in mind the facilitation of access to non-specialists of specific musical styles. One of the goals on the development of these applications is to open a new and potentially revolutionary way of education and active enculturation with unfamiliar musical styles. The user will be able to improve their skills in that style in an environment that can track how well she is performing, and give clues about how to improve the performance. As noted above, generative music applications are increasingly regarded as powerful tools for music education and performance [31, 32]. They have a high potential to motivate a re-discovery of music as a way of communication, compared to the currently dominating mode of consumption, in which digital devices are only used passively.

**Acknowledgements**

To George Sioros, Rui Dias, Konstantinos Trochidis, and Akshay Anantapadmanabhan for their commitment on the research projects that led to the creation of the aforementioned applications and for the brainstorms that led to the refinement of these ideas. The *CaMel* application was developed in project “Cross-disciplinary and multicultural perspectives on musical rhythm” funded through NYU Abu Dhabi Institute’s Research Enhancement Fund grant. The kin.rhythmicator, kin.recombinator, and *GimmeDaBlues* applications were developed as part of project "Kinetic controller, driven, adaptive and dynamic music composition systems" funded by the ERDF through the Program COMPETE, by the Portuguese Foundation for

[5] http://www.ixi-audio.net/content/download/ixiquarks/

Science and Technology (FCT), Project ref. FCOMP-01-0124-FEDER-011414, UTAustin/CD/0052/2008.

## 5. REFERENCES


[1] C. Guedes. “Composição em tempo real.” Unpublished typescript. Text submitted for support of public lesson for promotion to Associate Professor at the Polytechnic Institute of Porto, Portugal, 2008.

[2] J. Chadabe. “Interactive Composing.” *Computer Music Journal* vol. 8, no. 1, pp 22–27, 1984

[3] A. Eigenfeldt. “Real-time Composition or Computer Improvisation? A Composer’s Search for Intelligent Tools in Interactive Computer Music.” *Proceedings of the Electronic Music Studies*, 2007

[4] A. Eigenfeldt. “Real-time Composition as Performance Ecosystem.” *Organised Sound* vol. 16, no.2, pp. 145–153, 2011

[5] A. Brown. “Creative Partnerships with Technology: How Creativity Is Enhanced Through Interactions with Generative Computational Systems.” *Proceedings from the eighth Artificial Intelligence and Interactive Digital Entertainment Conference*, 2012

[6] O. Brown, A. Eigengeldt, P. Pasquier, & S. Dubnov (eds.). *Computers in Entertainment* vol. 4, nos 2 and 3. Special issue on musical metacreation part I/II, 2016

[7] R. Rowe. *Interactive music systems: Machine listening and composing*. Cambridge, MA: MIT Press, 1993

[8] R. Rowe. *Machine musicianship*. Cambridge, MA: MIT Press, 2001

[9] J. Chabade. 2002. “The limitations of mapping as a structural descriptive in electronic instruments.” *Proceedings of the New Instruments for Musical Expression Conference,* Dublin, 2002.

[10] T. Kvifte. “What is a musical instrument?” *Svensk tidskrift för musikforskning* vol. 90, no. 1, pp 45-56, 2008

[11] J. Drummond. “Understanding Interactive Systems”. *Organised Sound*, vol. 14, no. 2, 2009

[12] O. Brown, A. Eldridge and J. McCormack, 2009. “Understanding interaction in contemporary digital music: from instruments to behavioral objects.” *Organised Sound,* vol. 14, no. 2, pp. 188-196, 2009

[13] T. Winkler. *Composing interactive music: Techniques and ideas using Max*. Cambridge, MA: MIT Press, 1998

[14] Dodge, C., & Jerse, T. A. *Computer Music: Synthesis, Composition and Performance*. Macmillan Library Reference, 1985

[15] I. Xenakis. *Formalized Music: Thought and Mathematics in Music*. Indianapolis: Indiana University Press, 1992.

[16] H. K. Taube. *Notes from the metalevel: Introduction to algorithmic music composition*. London: Taylor & Francis, 2004.

[17] B. Eno, B. and P. Chilvers. *Bloom*. [Generative music application for iOS devices]. Opal Inc, 2008.

[18] C. Barlow. “Bus journey to Parametron: (all about Cogluotobusisletmesi).” *Feedback Papers*. Cologne: Feedback-Studio Verlag, 1980.

[19] C. Guedes. “Mapping movement to musical rhythm: A study in interactive dance.” Unpublished dissertation, NYU, 2005.

[20] C. Guedes. “Extracting Musically-Relevant Rhythmic Information from Dance Movemen by Applying Pitch-Tracking Techniques to a Video Signal.” *Proceedings of the Sound and Music Computing Conference*, Marseille, 2006, pp. 25-33

[21] C. Guedes. “Translating Dance Movement into Musical Rhythm in Real Time: New Possibilities for Computer-Mediated Collaboration in Interactive Dance Performance.” *Proceedings of The International Computer Music Conference*, Copenhagen, 2007

[22] C. Guedes. “Establishing a Musical Channel of Communication between Dancers and Musicians in Computer-Mediated Collaborations in Dance Performance.” *Proceedings of the Conference on New Interfaces for Musical Expression*, New York, 2007, pp. 417-418

[23] G. Sioros and C. Guedes..“A formal approach for high-level automatic rhythm generation.” *Proceedings of the BRIDGES 2011 Conference*. Coimbra, 2011.

[24] G. Sioros and C. Guedes.”Automatic Rhythmic Performance in Max/MSP: the kin.rhythmicator.” *Proceedings of the 11th International Conference on New Interfaces for Musical Expression*, Oslo, 2011.

[25] G. Sioros and C. Guedes. “Complexity-driven recombination of MIDI loops.” *Proceedings of the Conference of the International Society for Music Information Retrieval* Miami, 2011.

[26] C. Barlow. “Two essays on theory.” *Computer Music Journal,* vol. 11, pp. 44-60, 1987.

[27] R. Dias, T. Marques, G. Sioros and C. Guedes. “GimmeDaBlues: An intelligent Jazz/Blues player and comping generator for iOS.” *Proceedings of the Conference on Computer Music Modeling and Retrieval,* London, 2012.

[28] G. T. Toussaint. “Measuring the Perceptual Similarity of Middle-Eastern Rhythms: A Cross-Cultural Empirical Study.” *Proceedings of the Fourth International Conference on Analytical Approaches to World Music*, New York, 2016

[29] L. Van der Maaten and G. Hindon. “Visualizing data using t-SNE.” *Journal of Machine Learning Research* vol. 9, pp. 2579-2605, 2008.

[30] T. Iwai. *Electroplankton*. [Game for console Nintendo DS]. Nintendo, 2005

[31] A. Brown and S. Dillon. “Collaborative digital media performance with generative music systems.” In G. McPherson (Ed.), *Oxford handbook of music education, Volume 2*. (pp. 549-566). New York: Oxford UP, 2012.

[32] E. Tobias. “Let’s play! Learning music through video games and virtual worlds.” In G. McPherson (Ed.), *Oxford handbook of music education, Volume 2*. (pp. 531-548). New York: Oxford UP, 2012.