Drumming with style: From user needs to a working prototype

Sergi Jordà, Daniel Gómez-Marín, Ángel Faraldo, Perfecto Herrera
Music Technology Group
Roc Boronat, 138
Barcelona, Spain
name.surname@upf.edu

ABSTRACT

This paper documents and discusses the process of developing a generative drumming agent built from the results of an extensive survey carried out with electronic music producers. Following the techniques of user-centered interaction design, an international group of beat producers was reviewed on the possibility of using AI algorithms to help them in the beat production work-flow. The results of these tests were used as design requirements for constructing a system that would indeed perform some tasks alongside the producer. The first results of this working prototype, a stylistic drum generator that creates new rhythmic patterns after being trained with a collection of drum tracks, are presented with a description of the system. Further stages of development and potential algorithms are also discussed.

Author Keywords

User studies, survey, drumming, music production, smart agents, generative music, algorithmic composition, Markov chains.

ACM Classification

H.5.5 [Information Interfaces and Presentation] Sound and Music Computing, H.5.2 [Information Interfaces and Presentation] User Interfaces, J.5 [Computer Applications] Arts and Humanities

1. INTRODUCTION

There are many pieces of music software that have been solely developed by following the intuitions and needs of their own developers. Even intended-to-be universal music composition environments show the imprint of their creators' musical interests or biases. Commercial software, on the other hand, has to be developed according to the requirements of potential users. Listening to users, current or potential, is one of the elements in their recipe for success in software development. It is, nevertheless, rare that music creation prototypes that have arisen in academic environments include, in their inception phases, the insights provided by potential users.

Computer-assisted music creation epitomizes another distinction between the academic and commercial sides of music technology. While algorithmic composition and AI-based music systems have a long tradition in academic and experimental realms, such tools seem not to have found their place in the market yet. Is it really because users would reject them?

In this paper we report on the process, findings and consequences of poking into the conceptual and practical views of users, and of letting them test rough ideas with software prototypes yet-to-be-converted into working tools. From this exploration, our resulting prototype addresses the generation and variation of rhythm patterns by means of high-level control, and according to some stylistic conventions.

1.1 Helpers for Popular Music Creation

Much of popular music, and electronic dance music (EDM) makes no exception, arguably starts out by imitation, by recalling a musical sketch, be it a melody, a chord progression, or (as most commonly in EDM) a rhythmic pattern. Progressively changing this material and adding variations, may finally turn the music into an original piece of its own. Grote [13] discusses whether this selection of a structural framework from fragments of one own's cultural background is already a creative act, or is perhaps more akin to the tasks of a scientist collecting literature for a state-of-the-art in a new paper. Within this context, and considering the greater realm of cultural practice where the repetition of established patterns or sounds would appear as redundancy, the use of generative agents or helpers in commercial popular music creation would seem a natural move. As Grote also points out, if machines were able to provide you with proposals for ready-made structural frameworks of a piece of music that fit your cultural background, if they were able, in summary, to do all the basic work for you, why wouldn't you embrace the possibility to save time and be more productive?

We know that such potential tools are far from constituting a mainstream trend yet, but where are the main shortcomings? Do they come from the natural fear we humans (and probably more specially popular music artists) have, of becoming progressively unneeded and supplanted by machines? Are they caused by a strong sentiment against the idea of a machine taking control on the entire groundwork of a music piece? Or are they rather a consequence of the technical difficulties imposed by hypothetical musical know-hows, that (specially in the case of electronic music) have not been fully analyzed neither formalized, and are in permanent evolution (so much that a one year-old EDM piece might sound horribly dated to some critically trendy ears)? While we acknowledge that at its current stage, the research documented in this paper will not be able to provide us definite answers to these questions yet, they will guide us and inspire us for which directions to take.

NIME'16, July 11-15, 2016, Griffith University, Brisbane, Australia.

2. PRELIMINARY RESEARCH

2.1 First Interviews with Experts

With the aforementioned questions in mind, our preliminary inquiries started at the Red Bull Music Academy (RBMA) that took place in Tokyo, in November 2014. Begun in 1998, the RBMA is a global music institution that organizes world-traveling series of music workshops. Its main event is an annual workshop in which two dozens of promising and not-yet-popular young music producers and DJs, selected from more than 4,000 submissions from about 100 countries, gather for two weeks of lectures, master classes, studio production workshops and collaborative writing and recording sessions[1]. Within the GiantSteps project [16], 16 in-depth interviews were conducted in Tokyo with RBMA participants. Ages ranged from 21 to 35, with 10 users in the 20-30 age group and 6 older than 30. Each user was interviewed individually in one session that lasted between 30 and 120 minutes. The interviews were conducted in between RBMA work sessions and during recording studio down time, and focused on the musicians' work practices, their attitude towards the computer as a more or less active collaborator in their work process, as well as on their visions and concerns for the future, their imagined ideal tools and interface wishes. Interviews were recorded in audio, transcribed and subsequently analyzed. The results are fully documented in [2]. We here provide a very quick summary, concentrating on aspects that may orient our pursuit.

While the interviews reveal to a certain degree the musicians' fear of intelligent agents taking control of their music and making it less interesting, there is a common agreement that machines could provide help in many tasks, even in the ones considered more "creative". Most producers agreed that tools that would not impose their own aesthetics but would rather work within some stylistic constraints defined by the users, and that would specially preserve (unless explicitly stated otherwise) the musician's aesthetic stamps would be highly appreciated. According to the producers interviewed, such tools, which should also allow accidents or unexpected results when deliberately pushed beyond their declared limits, would be specially helpful for suggesting variations and adding variety to the users' material. Experts also call for tools allowing them to adopt a perspective that hides or silently cares for low-level details and non-creative stuff.

2.2 Three Conceptual Pillars

From these preliminary guidelines we approach the design and development of a stylistic drum generator that allows to interactively modify rhythmic patterns and create new ones, after being trained with a collection of drum tracks. We decide to start with rhythm, because it is one of the primary components of EDM, and we base our prototype on three conceptual pillars suggested by the interviews: (i) the provision for high-level control of the system's behavior (i.e., by means of domain-specific conceptual parameters), (ii) the embedding of style knowledge at the core of the musical representations, and (iii) the metaphor of "model plus variations" to create, in controllable and meaningful ways, output diversity.

In the following section we provide some justification for and clarifications about each of these, together with an overview of existing related approaches.

[1] http://www.redbullmusicacademy.com/

3. RELATED APPROACHES

3.1 High-Level Control

In computer music jargon, "high-level control" is an idea that goes back to the 1970's if not earlier, and which relates to the concepts of "Interactive Music" introduced by Chadabe (1984), and to Laurie Spiegel's "Intelligent Instruments" (Spiegel, 1987). For Pressing (1990), performing with such musical systems can be closer to "conducting an orchestra" than to what we conventionally understand by "playing a musical instrument". While this approach has been ubiquitous for at least two decades in more experimental electronic music, and can indeed be seen as the backbone of much laptop performance, which often relies on real-time programming environments such as Max/MSP, Pure Data or SuperCollider, it is still not so prevalent in more mainstream electronic music, neither in the commercially available tools typically used within these styles.

3.2 Interacting with Style

The notion of musical style as a statistical model dates back to 1956, after Leonard Meyer's seminal work Emotion and Meaning in Music [18], partially motivated by Shannon's Mathematical Theory of Communication [22]. Whilst there have been approaches to style-modeling based on formal grammars ([24] specifically for jazz harmony) and the implementation of music-theoretical rules [11], we consider statistical modeling of style a fruitful approach toward generative music systems ([9]; [8]; [21]), especially for the genres and subgenres we are dealing with, about which there is not a formalized musical theory proper.

Electronic dance music seems to be deeply anchored in conventions that define the different genres and subgenres, emerging from a complex network of factors, including aspects as diverse as marketing trends and social stratification, technological development and studio production techniques, not to mention the different musical roots from diverse ethnicities and geographical idiosyncrasies. However, we will not consider these aspects inasmuch as they are highly debatable and vague [25], focusing instead on style, in the tradition set by Meyer, referring to those aspects that can be statistically learnt from a given collection, however heterogeneous this might be (e.g. a user's music collection). In that sense, we hope to alleviate the danger of dealing with preconceived ideas about genre and the lack of (or blind confidence in) ground-truth [26].

When considering style imitation, Markovian sequence generation is a well known modeling tool, which has been used in many generative musical applications ([1]; [6]; [11]; [19]). More specifically, Markov chains have also been applied in real-time interactive music systems, such as "M" [34], the "Continuator" [20] or "Omax" [10]. However, Markov chains and interactive control are two concepts that do not go well together, because a user may not be able to specify additional musical properties wished in the generated material, while preserving Markovian properties and therefore stylistic consistence (Pachet, 2011). Pachet proposes the use of Elementary Markov Constraints (EMC) as a computational solution for obtaining steerable or interactive Markovian sequences. In our prototype, we explore complementary approaches based on the constrained and perceptually meaningful interactive manipulations of Markov probability tables.

3.3 Source/Model and Variations

Our third pillar, based on the idea of applying variation to a model or source chosen by the musician, seems also safer than the alternative suggestion of material from scratch, for the artist who wants to keep full control over her creations (even when this material complies with the stylistic constraints defined in the previous section). This idea of "controlled variation" should not prevent, however, the suggestion of "weird" or contrasting material, when deliberately decided by the musician.

We conclude this section with an overview of existing algorithmic systems for drum pattern generation, both in the academic and in the commercial realms.

3.4 Algorithmic Systems for Drum Pattern Generation

Most of the systems reported in the literature try to capture the style of a compilation of patterns and embed it in original rhythms that resemble those of the compilation. Focusing on the input side, these systems can be classified in two groups, according to the music representation they are based upon: symbolic notation versus audio. Independently of the input format, the techniques used for the analysis and synthesis of the rhythms are diverse, being genetic algorithms (GA), neural networks and stochastic processes the most commonplace.

On the first group of symbolic input based systems, Burton's system uses a GA to recombine collections of polyphonic drum patterns extracted from drum machines and transcribed manually [7]. As a part of the GESMI project, aimed at generating complete electronic music tracks, Eigenfeldt uses 1st order Markov chains of 32 steps resulting from the analysis of the drum tracks of 100 transcribed electronic music songs [12]. Tidemann et al. present a system based on Echo State Networks (ESN), a particular approach of a neural network that is trained in real-time by a human MIDI drummer [28]. Once their system is trained, it is set to imitate the sequence that had been used in training. Bernardes et al. use a GA to create new polyphonic drum patterns, based on the study of a set of MIDI drum loops. The main operations of a GA which provide a variable population of rhythms are crossover and mutation. Crossover is based on a first order Markov chain and mutation on the selection of a step to transform controlled by their metrical weights. Once a population is created, density and complexity are used as user inputs to filter out the output drum patterns [5].

On the second category we find two audio-based drumming systems. Aucouturier and Pachet [4] describe a reactive system that adapts to the musical input of a performer on a MIDI keyboard. The generative system is based on extraction of drum sounds from recordings and then uses concatenative synthesis to generate rhythms. In the example reported, the mappings between MIDI and drum generation, as well as other generative controls and constraints are defined offline, therefore letting the system to drum along with no real-time control. Wooler [33] describes a fast adaptive system used to create rhythm mosaics resulting from two audio sequences to be cross-faded at the user's will. Cross-fades are not for volume but for the percussive elements extracted from one track or the other and located in non-disruptive positions. The rules for locating the fragments are based on a Markov analysis of the short rhythms to be cross-faded. These two last examples are borderline stylistic, but are worth mention due to the creative approach to polyphonic rhythm generation.

Most commercial drum programs and plugins available are concerned with sound rendering (synthesis and sampling) and basic sequencing rather than with intelligent pattern generation or algorithmic composition. We present here a brief summary of the most relevant programs we have found connected to our research.

Different Drummer [27] and *Robotic Drums* [30] use stochastic methods for generating drums. Both are drum sequencers in which events on a given step are user-controlled by a probability value. Another approach is *Stylus RMX* [23], which aims to create music tracks based on overlapping pre-analysed audio samples, forcing the onsets to be displaced to certain points in the grid. There are two variation parameters: a "simplify" knob, which reduces the amount of onsets in the loop, and a discrete selection menu called "variation", where a fixed amount of variations from the original patterns can be selected. Although not a drum-exclusive application, drum loops can be loaded in order to be transformed.

Electronic artist Cristian Vogel has applied the Euclidean algorithm [29] to automatic pattern generation in the Kyma environment [31] and he has used the software to create all the rhythmic elements for the 2014 album "Polyphonic Beings". WaveDNA has recently released *Liquid Music* [32] for Max for Live, which provides building blocks of rhythmic patterns that can be varied and tweaked with unique visual editing tools such as the "beatform tumbler" complexity transformer, the "beatform weaver" combination generator or the "groovemover" remixer. Artist James Holden has tackled the difficult notion of groove and the challenges that need to be addressed when interacting with human musicians. Based on Holger Hennig's ideas, who examined the effects of synchronisation between musicians [14], he has released a free MIDI humaniser [15] which can listen and respond to musicians in real-time performances.

4. REFINING THE DESIGN

4.1 Drum Interaction Questionnaire

For deepening into the potential high-level user control features of our drum pattern creation software, we subsequently conducted an online questionnaire. It consisted of 25 questions, all using a 5-point Likert scale, which were grouped according to the following topics: the musical expertise of the participant (questions 1-4), the concepts of style-knowledge and variations (questions 5-10), the potential interest of such systems namely in an off-line composition environment and in a real-time performance context (questions 11-12), the hypothetical and interesting properties, features, control parameters, etc. such a system could potentially have (question 13), the specific interest of 5 potential controls, namely: density, syncopation, stylistic typicality, variation frequency and variation amount (questions 14-19), the convenience of having individual controls per voice (question 20, 21), and the desired input and output possibilities of such a system, namely audio and MIDI (questions 22-25).

A total of 48 participants took part in the survey, of which 38 were practicing musicians, 25 played drums or percussion, and 36 produced electronic music and programmed drums patterns. Following we present and discuss some of our findings:

4.1.1 Style and variations

All features related to style and variations were considered very interesting, the more appreciated being the possibility for users to create their own styles (4.40)[2] and the least one, the possibility to navigate between different styles (3.90) (perhaps some participants might have found this feature harder to grasp). The possibilities to work with a model or source pattern to which to apply variations and to be able

[2]All subsequent values indicate the average value of the ratings, from a maximum score of 5.

to navigate among several of these patterns were both also very much appreciated (4.35).

When asked about the hypothetical use of these agents in a *real-time performance and off-line music composition tasks*, more participants thought that these potential systems could be more helpful for off-line composition tasks than for real-time performance (4.23 vs. 3.83). This tendency seems very reasonable considering that participants could not evaluate the quality of the outcomes yet. We consider, though, that these results still show a quite positive predisposition for future live experimentation with such devices.

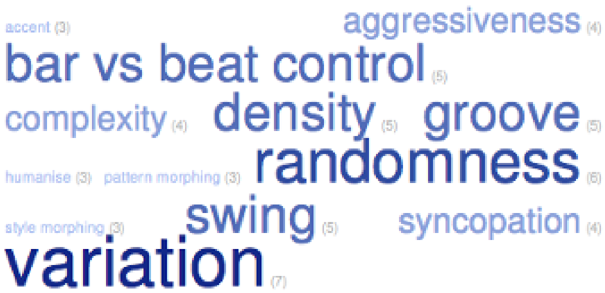

Figure 1: More popular concepts with size proportional to their respective frequency of appearance.

4.1.2 Open suggestions and tags

Participants were asked to list any potentially interesting high-level interactive control parameters that such systems could afford (within this question, they were instructed about what could be considered a "high-level" control). Figure 1 shows the more frequent tags, from a list of manually post-processed 118 terms that comprised all the 48 participants' contributions. We observe that the tags that repeat more frequently are "variation" (7) and "randomness" (6), followed by "bar vs. beat control", "density", "groove", "swing" (5), "aggressiveness", "complexity", "syncopation" (4), and finally "accent", "humanize", "pattern-morphing" and "style morphing" (3).

4.1.3 Density, Syncopation, Stylistic typicality, Variation rate and Variation amount

Following this open-form suggestion, we explicitly asked participants for the convenience of some specific controls we had previously identified ourselves as potentially interesting and reasonably implementable. All these concepts had already frequently appeared in the open tags list, either explicitly (i.e. density and syncopation), or via related concepts (e.g. commonness, weirdness, commonness-weirdness, absurdity and contrast for stylistic typicality, and variability, variation, variation-amount or bar-vs-beat-control for variation frequency and variation amount). The topics we introduced were namely *density*, *syncopation*, *stylistic typicality*, *variation rate* and *variation amount*[3]. All features were considered quite interesting, confirming the results obtained in the open list, with syncopation being considered the most interesting one (4.29), and variation rate as the least (3.69).

4.1.4 Global vs. individual (voice) control

The issue of a global control of the drum pattern vs. a separated control for each drum voice (i.e. kick, snare, etc.) had already appeared in the open list form, with related tags such as: density by voice, voice separated control, voice hierarchy, voice number control or polyrhythmic support. We were now posing these two precise questions: "For each of these parameters I would like to have global direct control on all the sounds (e.g. kick, snare, open hat, closed hat...) at once" and "I would like to have separate control on each of the sounds (e.g. kick, snare, open hat, closed hat...)". Perhaps not surprisingly, both possibilities were considered equally important (3.96 vs. 3.98, respectively).

4.1.5 MIDI and Audio Input/Output

As an introduction to the two last questions, we uncovered and explicitly stated the idea that such a system would be trained in a "style" by being fed with examples (e.g. from individual files or from complete folders). These questions thus addressed the preferred formats for this training material (either audio or MIDI drum loops), as well as the preferred output formats (audio or MIDI messages). Again, the preferences were very similar and all features were considered especially important (4.1 for MIDI importing to 4.45 for MIDI output).

As we understand it, an ideal system should be able to work in all modalities. It should be able to learn from MIDI sequences (definitely simpler to implement), but also from audio drum loops (which should probably be decoded before analysis). It should be able to output MIDI messages, a flexible solution for the professional producer who will definitely want to have full control over the chosen drum sounds, but it should also be able to produce audio, probably resulting from the advanced manipulation of the drum loops used as models or sources.

4.2 Prototype Requirements

The preliminary RBMA interviews together with the results of this drum interaction questionnaire led to specific concepts and ideas, which were translated into requirements and set to work in the design process of a novel drum interaction system:*Drumming with Style*. This prototype is an interactive drum pattern generator/variation generator, based on Markov chains, that pursues the concept of interacting (or high-level "drumming") with style. Following, we describe its current state, with a focus on its interactive control aspects.

5. DESCRIPTION OF THE SYSTEM

Drumming with Style is a software prototype for both performance and studio work, aimed at the generation and variation of drum patterns. It allows continuous pattern variations to be controlled by the performer on the basis of some high-level musical parameters, such as density, syncopation and commonness, while keeping compliance with a user-defined drumming style. At its present state, the material used to feed its stylistic knowledge are collections of MIDI drum loops or drum tracks, which can be selected by the user. In its current incarnation it is implemented in Pure Data and can be run as a VST plugin using PdPulp[4], thus working in synchronization with any DAW. It outputs MIDI messages that can be sent to the same DAW or to any drum synthesizer/sampler plugin.

5.1 Step-Based Markov Probability Tables

The material used to feed the stylistic knowledge to our prototype are sets of MIDI drum loops or complete drum tracks. Each set is assumed to contain genre-style-consistent information (e.g., *deep-house*, *jungle* drum patterns, or any collection compiled by the user). In the current

[3]Before each question, each of these terms was briefly described in a short sentence

[4]http://pd-pulp.net/

version we are working with sequences of up to eight different instruments (i.e. kick, snare, open hat, closed hat, rim shot, clap, bongo and conga), but the same principle could be extended to drum sequences with more instruments.

Each MIDI library or set of tracks selected as the stylistic input is analyzed, filling several probability distribution tables depending on the selected order. The current implementation works with Markov orders from 0 to 8. The resulting tables are n 2D-arrays $Anij$, $i \in [1-256]$ and $j \in [1-32]$, where n represents the Markov order used, j represents the step in the drum loop and i represents the possible drumming events at step j. The 256 possible values of i cover silence (no instrument sounding), each instrument sounding separately , plus all the possible instrument combinations. As the Markov order n increases, so does the amount of probability distributions. For example, if the Markov order is 2 (n=2) and $i \in [1-256]$, at step *j-2* we have 256 possible events and at step *j-1* we have another 256. In general, the amount of tables increases at a rate of i^n, so that $256^2 = 65536$, is the number of tables needed to record all possible combinations of two past events.

5.2 Interaction Parameters

At the heart of a Markov system lays a selection of possible future events mediated by a probability distribution table which is selected given the past events. Our system enables the user to modify in real-time these probability tables based on high-level controls that will favor (or disfavor) the occurrence of some events at given steps. The following sections describe these high-level concepts and their implementation into interactive algorithms. Namely, we currently have addressed the concepts of *commonness/oddness*, *density* and *syncopation*, which are derived from the interviews and from the questionnaire with producers.

5.2.1 Strategies for Commonness/Oddness

The notion of *commonness* relates to the manipulation of the probability of an event to occur at a given step. This opens up the door to increasing the probability of the most probable/common events, at the cost of reducing the probability of the least common ones. Stylistically, this could be translated as emphasizing the most recurrent elements in the database, or as "sharpening" the style. On the other hand, the concept of *oddness* suggests the opposite: emphasizing the least recurrent drumming events found in the database (the odd rhythms within a style)[3]. So far we have implemented this "commonness/oddness" control using a sigmoid transfer function, which works as follows: probability distributions are temporally reordered at every given step, according to their probability values, and subsequently multiplied by the sigmoid, whose skewness and slope [-1,+1] are controlled by the user. This algorithm allows to sharpen or flatten these probabilities (via the *skewness* control), but also to invert them (with the *slope* control), thus creating a sort of "anti-style".

5.2.2 Strategies for Density

Density is defined as the amount of onsets of a given instrument per unit of time. We are currently exploring density in two ways. First, as the balance between silence and the occurrence of an event, and second, as the balance between silence and the different amount of drum sounds played together. On the first approach, the density of a pattern on a given step is inversely proportional to the probability of a "silence" event. For example, when the probability of reproducing a silence on a given step is 100%, then that step has density 0. Controlling the density of a pattern thus becomes controlling the probability of silence to occur at the expense of all the other 255 possible events. On our second approach to density, combos of simultaneous drum shots are considered to have a higher density than combos of lower amount of simultaneous drum shots (i.e. a combo of kick + snare + open high hat is denser than a combo of snare + clap). When manipulating density using this second approach, what is taken from (or given to) silence is distributed among (or taken from) the rest of the possible events, proportionally to their amount of simultaneous shots.

When density implies artificially increasing the value of an event on the probability distribution from zero to any other value (i.e. the occurrence of silence at a given step is 100% but we want to introduce a sound), it creates a rupture in the flow of a Markov system. This inconsistency is due to the appearance of new events that did not occur when the style was analyzed, that make the system unable to respond. This inconsistency is called a zero frequency state (ZFS). A ZFS is mathematically induced by an algorithm adding to a probability table instead of multiplying, since a probability zero multiplied by any value will always remain null. Our density algorithm takes care of these issues, by avoiding the use of zero values in the frequency tables and replacing them by very small values close to zero, and by automatically reducing the order of the Markov table to 0, whenever a ZFS is found. These steps add flexibility to the system allowing the increase of the probability value of any given step, regardless it not occurring on that step in the database.

5.2.3 Strategies for Syncopation

Our approach to syncopation is based on the metrical weight profile by Lerdahl and Jackendoff [17], where all the steps within a rhythmic pattern are assigned a weight according to the metrical structure of the rhythm. This profile is said to describe human cognitive expectation of an onset when listening to a 4/4 beat. The higher the metrical weight, the higher are our expectations of experiencing a note on such step, given the metric and the resolution. When the expectations are broken, such that an onset that was expected to occur on a strong metrical weight is presented on a previous step, or that a strong metrical step is replaced by a silence, a syncopation occurs.

With the syncopation control, a user is able to amplify the syncopation of a pattern, or on the contrary, to make the accentuation of the beat stronger. Our syncopation algorithm changes the density value at a given step, while excerpting the same value on the next step in opposite direction. If the intention is to increase syncopation, the odd steps are density-increased and the even ones decreased. The opposite happens if the intention is to reduce syncopation, giving more density to the steps that reinforce the sensation of the beat. In general terms, syncopation is a step-dependent density control.

6. FUTURE WORK

We are still adding features to the system. An *Independent instrument control* will allow for separate controls over the different drum sounds, such as increasing the density of the kick while decreasing the snare's. With *Style-to-model interpolation* a source pattern is input by a user, and variations of it are generated by increasing the statistical influence of a style. In *Style-to-style interpolation*, a pattern is generated somewhere in between two or more different styles, according to an interpolation value that indicates the statistical influence of each style in the resulting pattern. Other controls described in the reported questionnaire, such as the variation rate and the variation amount, have neither been addressed yet. More importantly, although we are frequently conducting many informal local tests, specially for addressing the choice and implementation of some algorithms, all of the current solutions and alternative approaches we are implementing still need to be further validated by the users.

7. CONCLUSIONS

We have presented *Drumming with Style*, a musical agent for the interactive generation and variations of drum patterns, according to some stylistic constraints defined by the user. The guidelines for this agent have been deduced from extensive interviews and questionnaires carried out with dozens of EDM producers. This is a work in progress; all ideas of interaction presented in this paper have been implemented, and are being tested with users and explored more in depth, in order to understand which of them, and under which circumstances, are the most suited for our system.

8. ACKNOWLEDGMENTS

This research has been partially supported by the EU funded GiantSteps project[5] (FP7-ICT-2013-10 Grant agreement nr 610591).

9. REFERENCES

[1] C. Ames. The Markov process as a compositional model: a survey and tutorial. *Leonardo*, 22(2):175–187, 1989.

[2] K. Andersen and F. Grote. Giantsteps: Semi-structured conversations with musicians. In *Proceedings of the 33rd Annual ACM Conference Extended Abstracts on Human Factors in Computing Systems*, CHI EA '15, pages 2295–2300, New York, NY, USA, 2015. ACM.

[3] K. Andersen and P. Knees. The dial: Exploring computational strangeness. In *Proceedings of the 34rd Annual ACM Conference Extended Abstracts on Human Factors in Computing Systems (CHI EA '16)*. CHI, 2016.

[4] J.-J. Aucouturier and F. Pachet. Ringomatic: A Real-Time Interactive Drummer Using Constraint-Satisfaction and Drum Sound Descriptors. In *Proceedings of the International Conference on Music Information Retrieval*, pages 412–419, 2005.

[5] G. Bernardes, C. Guedes, and B. Pennycook. Style Emulation of Drum Patterns By Means of Evolutionary Methods and Sta- Tistical Analysis. In *Proceedings of the Sound and Music Conference*, pages 1–4, 2010.

[6] F. Brooks, A. Hopkins, P. Neumann, and W. Wright. *An experiment in musical composition*. Cambridge: MIT, 1993.

[7] A. R. Burton. *A Hybrid Neuro-Genetic Pattern Evolution System Applied to Musical Composition*. PhD thesis, 1998.

[8] D. Conklin. Music Generation from Statistical Models. *Proceedings of the AISB 2003 Symposium on Artificial Intelligence and Creativity in the Arts and Sciences*, pages 30–35, 2003.

[9] D. Conklin and I. H. Witten. Multiple viewpoint systems for music prediction. *Journal of New Music Research*, 24(1):51–73, 1995.

[10] A. Cont, S. Dubnov, and G. Assayag. Anticipatory Model of Musical Style Imitation Using Collaborative and Competitive Reinforcement Learning. In *Anticipatory Behavior in Adaptive Learning Systems*, pages 285–306. 2007.

[11] D. Cope. *Virtual music: computer synthesis of musical style*. MIT press, 2004.

[12] A. Eigenfeldt and P. Pasquier. Evolving Structures for Electronic Dance Music. In *Gecco'13: Proceedings of the 2013 Genetic and Evolutionary Computation Conference*, pages 319–326, 2013.

[13] F. Grote. The Music of Machines: Investigating Culture and Technology in Musical Creativity. In *Culture and Computer Science*, pages 219 – 230, 2015.

[14] H. Hennig. Synchronization in human musical rhythms and mutually interacting complex systems. *Proceedings of the National Academy of Sciences of the United States of America*, 2014:1–6, 2014.

[15] J. Holden. James Holden: On Human Timing, 2015.

[16] P. Knees, K. Andersen, S. Jorda, M. Hlatky, G. Geiger, W. Gaebele, and R. Kaurson. Giantsteps-progress towards developing intelligent and collaborative interfaces for music production and performance. In *Multimedia & Expo Workshops (ICMEW), 2015 IEEE International Conference on*, pages 1–4. IEEE, 2015.

[17] F. Lerdahl and R. Jackendoff. *A generative theory of tonal music*. MIT Press, 1985.

[18] L. B. Meyer. *Emotion and meaning in music*. University of chicago Press, 2008.

[19] G. Nierhaus. *Algorithmic composition: Paradigms of automated music generation*. 2009.

[20] F. Pachet. The Continuator: Musical Interaction With Style. In *Proceedings of the International Computer Music Conference*, pages 333–341, 2002.

[21] M. Pearce, D. Conklin, and G. Wiggins. Methods for Combining Statistical Models of Music. *Computer Music Modeling and Retrieval*, 3310:295–312, 2005.

[22] C. E. Shannon. A mathematical theory of communication. *The Bell System Technical Journal*, 27(July 1928):379–423, 1948.

[23] Spectrasonics. Stylus RMX, 2005.

[24] M. J. Steedman. A generative grammar for jazz chord sequences. *Music Perception*, pages 52–77, 1984.

[25] B. L. Sturm. A survey of evaluation in music genre reconigtion. In *Adaptive Multimedia Retrieval: Semantics, Context, and Adaptation*, volume 8382, pages 29–66. 2014.

[26] B. L. Sturm and N. Collins. Four Challenges for Music Information Retrieval Researchers. *Digital Music Research Network 9*, 2015.

[27] Technemedia. Different Drummer, 2015.

[28] A. Tidemann, O. PinarÌĹurk, and Y. Demiris. A Groovy Virtual Drumming Agent. volume 5773, pages 104–117, 2009.

[29] G. Toussaint. The Euclidean Algorithm Generates Traditional Musical Rhythms. *BRIDGES: Mathematical Connections in Art, Music and Science*, pages 1–25, 2005.

[30] H. Urtubia. Robotic Drums, 2015.

[31] C. Vogel. Rhythmic Computation Lab, 2015.

[32] Wavedna. No Title, 2015.

[33] R. Wooller and A. R. Brown. Note sequence morphing algorithms for performance of electronic dance music. *Digital Creativity*, 22(1):13–25, 2011.

[34] D. Zicarelli. M and Jam Factory. *Computer Music journal*, dec(1):13 – 29, 1987.

[5] http://www.giantsteps-project.eu