Rhythmically-Controlled Automata Applied to Musical Improvisation

by
Sergio Krakowski Costa Rego
Submitted to the Computer Graphics Department
in partial fullfilment of the requirements for the degree of
Doctor in Philosophy
at the
Instituto Nacional de Matemática Pura e Aplicada
Rio de Janeiro

Advisor: Luiz Velho
Co-Advisor: François Pachet

30th september 2009

Contents

List of Figures

To Fayga Ostrower (*in memorian*),
who believed there is something in-between *art* and *science*.

Abstract

The main contribution of this thesis is to propose a Framework to generate Rhythmically-Controlled Multi-Mode Audio-Based Interactive Systems used to create variable-form improvised pieces in real musical performance contexts. In the recent years the field of Interactive Music System has grown considerably but issues related to Rhythm have been underestimated. This thesis starts with a comparison between several Interactive Music Systems related to the above mentioned problem. After that, A profound analysis with respect to Rhythm is developed considering different perspectives. In the core of this work, the concepts of mode and meta-mode of interaction are presented and formalized using the Theory of Synchronized Automata empowered by Tree Structures. Based on this formalization, some computational tools are developed to deal with real life situations. The two case studies built using these tools prove the approach here presented do solve the central question posed by this thesis. They are recorded on video, described and analyzed in details in this thesis.

Aknowledgements

Grazie Cristina, per avere fatto in modo che le parole non siano sufficienti per ringraziarti.

Obrigado ao Renato dos Santos.

Obrigado a todos os meus amigos que me suportaram nesta batalha. Incluo aqui minha mãe, Lilian, e meu pai, Osvaldo, assim como meus avós David e Nicha, e toda a minha família.

Obrigado ao Luiz Velho, que aceitou esse desafio e me deu todo o suporte necessário. Obrigado também a Paulo Cezar.

Je remercie Francois Pachet, Jean-Pierre Briot, Pierre Roy et tous de Sony/CSL-Paris.

Obrigado Hedlena e Giordano! Mercie, Tristan!

Agradeço também a Zé Guaranys e Marcos Campello.

Finalmente, agradeço aos músicos que me ajudaram a manter a esperança.

E finda a espera.

Chapter 1

Introduction

In 2003, the group I played with, named Tira Poeira, has gained a certain projection in the Brazilian media channels as being the group which represented the renewal of the Choro, genre of instrumental music born in the mid-XIXth century considered the grandfather of Samba.

As we considered ourselves a "modern" group that have mixed many diverse influences to that style, we felt the inevitable obligation of experimenting with "electronic music", whatever that could mean to us. We completely ignored the field of Computer Music and the only "technology" available to us was to synthesize a track full of "cool and modern" sounds and loops, record it in a CD, push the play button and play over it.

The genesis of Tira Poeira came from a common will: the search for freedom through improvisation, using, as our root and essence, the Choro. A typical Tira Poeira execution of a Choro piece included, after the theme exposition, several improvisation sections using jazz-like elements such as rhythmic illusions, reharmonizations, dynamically exploding solos, linked through unison complex bridges and followed by the re-exposition of the theme.

So, in that traumatic night we pushed the play button, our search for freedom was ruined, all the spontaneity of complex improvisation was lost, we went out of sync with the CD player, and our premiere mixing Choro and "electronic music" was, let's say, a bit frustrating if not a complete failure.

At least one good thing happened because of that night. I asked myself: is there a better way of "playing with machines"?

As a first step I decided to investigate how could I "play with the computer" using the Pandeiro, a Brazilian Tambourine I play professionally for fifteen years. This thesis is a result of that investigation.

First of all, I must let clear that all the choices made during this research are based on my own aesthetical interests and my own research as an artist. I believe any scientific work that deals with art has in it genesis and in its development a strong interference of the aesthetical choices of its creator. Nevertheless, the discussions and the results generated in this work are objectively discribed using mathematical tools and a clear problem is posed and solved. This makes this thesis scientifically valid with the advantage that its results usable in real life situation.

A first choice had to be made: whether to consider the computer another player or an

instrument. I chose the second option. This research do not intend to replace musicians by computers, but to extend musicians' capabilities.

The way this work unfolded was not linear. Although the problem I was trying to solve was not completely clear in the beginning, the approach found in many scientific sources seemed to focus on *modelling* reality instead of *effectively dealing* with it. As I was interested on the latter approach, and following the suggestion of my advisors, I decided to develop a series of experiments, some restricted to simple rhythmic interaction situations, others directed to the construction of entire musical pieces, but all of them with the common objective of providing a natural way of interacting with the computer through Music.

As a result of these experiences, I could formulate my problem in the following way:

How to let a musician build improvised solo pieces (about 3 to 5-minute long) using his/her instrument in interaction with a computer in a real life 'on stage' situation without loosing the spontaneity intrinsic to improvisation.

I now analyze, in an informal way, the main concepts and the approach used to solve this problem.

As the etymology of the word indicates, improvisation can be seen as the search for the unforeseen. In a solo situation, this is done through a narrative strategy where the musician have to find the right dose between repetition and variation if the goal is to capture the audience's attention ([Pachet 00]). An effective way of doing so is by breaking down this narrative (about 3 to 5-minute long) in smaller parts (about 30 to 50-second long) that have their own identity. When these parts are put together, the performer can create a *form* diversity that drives the listener's attention throughout the piece.

As I am interested in interactive situations, a digression must be done. We can informally classify the interaction between musician and computer in two groups: unpredictable and ideally predictable.

In the former case, the musician cannot completely predict what will be the computer's *musical answer*. This can happen when, for example, the computer uses a random variable to choose one of the parameters of its answer or when its behaviour is so complex that the musician looses the causal connection between his/her *musical act* and its response.

In the latter case, the musician expects a certain answer to his/her act and this is not satisfied only if the computer wrongly *interprets* that act.

Relating to what I mentioned before, in this search for the unforeseen, unpredictability is a central element.

In my research I delegate the responsibility of being *unpredictable*, *surprising*, or *creative*, to the musician, and not to interaction itself.

I chose to build several interaction *games*, some more predictable, others less, and let the performer choose when to use them, as building blocks of that narrative flow. They work as the parts of the improvisation solo and create that *form* diversity I was looking for.

As I mentioned before, the tone of our discussion until now is quite informal. The terms emphasized in this chapter that have an importance throughout this thesis will be reviewed and

rigorously defined.

I am now in a position to re-state both the problem and my solution to it, using a more rigorous "one sentence definition" that I dissect subsequently.
The use of Rhythmically-Controlled Multi-Mode Audio-Based Interactive Music Systems modeled by Synchronized Automata Theory augmented with Tree Structures bring a solution to create variable-form steady-tempo improvised solo pieces (about 3 to 5-minute long) using the Pandeiro.

Before starting our analysis, I shall make a small methodological digression. The strategy I adopt in this thesis is the following:

1. To build a *Mathematical Framework* to deal with the above problem,
2. Extract *Interactive Systems* as instantiations of this Framework, and
3. Use these Systems to create *Improvised Solo Pieces* that are videotaped and analyzed as a prove of the effectiveness of this approach.

The terms *Framework* and *Systems* are indiscriminately used throughout this text.

Coming back to our "dissection", first I shall aesthetically contextualize this research. The reason I focused on steady-tempo improvised solo pieces is an aesthetic choice based on my historical background. In Brazilian Instrumental Music such as in many subgenres of Jazz, the steady-tempo improvisation is a central element. I intend to build pieces aesthetically based in these two main references. Of course the generalizations to other contexts are welcome, but are outside of the scope of this thesis.

The systems are designed to work with the Pandeiro, but as they are Audio-Based, a possible generalization to other instruments can be done, as I will see later on.

As I mentioned above, I propose the idea of creating small *games* which are the building blocks of that narrative flow. These *games* will receive the name of *modes of interaction* and their interactive design is simple. That is why I chose the simplest tools of Automata Theory to model them.

I needed to put several of these modes together, in a Multi-Mode Interactive System, and I realized Synchronized Automata Theory was necessary to mathematically formalize that procedure. This Theory was yet insufficient since an hierarchical structure was missing.

I propose in this thesis the *Automata Tree*, a Synchronized Automaton augmented with a Tree Structure, to model Multi-Mode Interactive Music Systems. In a brief description we can see an Automama Tree as a complex Automaton that models the behaviour of an Interactive Music System. Its states model the way in which the computer will build its musical responses. The foundations and mathematical formalization of this approach are detailed further on.

To understand what is meant by a Rhythmically-Controlled System we shall consider it as that complex Automaton. My proposition is to use the rhythmic information of the musical audio signal generated by the musician as actions for this Automaton to change from one state to the other.

The inspiration for this approach came from an ancient percussion ensemble: The Bat Drum trio.

This group of three percussionists have a central role in the *Santeria* religion, "developed in 19^{th}-century colonial Cuba, by syncretized elements of Catholicism with the Yoruba worship of *orisha*" ([Schweitzer 03]).

The *Bat* drum trio is composed by a hierarchy of three drums, the *Okonkolo* (smaller drum), the *Itotele* (medium drum) and the *Yia* (biggest leading drum).

In a religious ceremony that constitutes this cultural context, the trio plays different percussion "small pieces" to each *orisha* deity that are concatenated by the *Yia* drummer. He does that by playing a "special call", a "rhythmic command" that is interpreted by the other two drummers as a sign to change for the next "small piece". During these "smaller pieces", the *Yia* can also make some "calls" that are intrinsic to that piece and are "answered" by the other percussionists in a certain specific way (see [Schweitzer 03] for further detail).

The analogy to my research is evident. The modes of interaction can be seen as those "small pieces" that are controlled by rhythmic phrases with the objective of creating a musical narrative.

There are some advantages on using rhythmic information as commands.

1. Rhythmic information is intrinsic to whatever musical signal,
2. Extracting this information from the digitalized audio is computationally cheap,
3. It lets the musician control the machine without stopping the musical flow, and
4. It is suited for timely precise interaction situations.

As we will see further on, these commands will be divided into two classes: *normal commands* used by the musician while playing with the modes of interaction, and *meta-commands* used by the musician to switch from one mode to the other.

The use of audio as the only input information in an interaction situation is one of my premises. I locate this research in the field of Interactive Music Systems, and not in the field of Musical Interfaces.

Another example of audio-based "musical command" are the phrases used by the saxophone player Steve Coleman to give orders to his band. This incredible musician developed with his group a vocabulary of some "phrase-orders". When he plays them, in whatever moment of the tune, his band has to follow him in a bridge and rearrange the way they interact with him by changing the tempo and/or the groove. Although this is not formalized in any article, it can be heard in the piece "Collective Meditation (suite)" from the album "The Tao of Mad Phat" available online at [Coleman Web].

An important fact worth noticing is that both the Bata and the Steve Coleman examples present a Sequential-Mode paradigm. In fact, when the *Yia* plays his "special call", it is interpreted as "go to the *next* part" of the entire piece, so as happens with Steve's "musical commands" that take his band into a "bridge part". The *Framework* I present is not of Sequential-Modes, but of Multi-Modes, which can be chosen whenever the performer wants, through *Rhythmic Phrases*, giving him/her complete freedom 'on stage'.

The structure of this thesis is the following:

In chapter two I present, first, a comparison between my systems and four other Interactive Music Systems. Then I make a small discussion about Music Interface approaches and their affinity to mine. Afterwards I give a small overview on other Accompaniment and Improvisational systems and close the chapter by discussing issues related to percussion and specifically to the Pandeiro.

When reading the scientific literature about Rhythm, I took a while to find an approach that aligns with my view on the subject, considering my historical background as a professional percussionist. Chapter three is devoted to an analysis of four different perspectives used to deal with Rhythm. Although the results obtained from this analysis are not directly used in the implementation of my systems, my work was only possible because my point of view differs from the accepted "mainstream" one and I hope the discussion in chapter three can be useful to inspire other different approaches.

Chapter four is the core of my thesis. There I present the definitions of *Mode* and *Meta-Mode* of Interaction that were the result of discussions I had with François Pachet. These definitions are done first informally, and then mathematically rigorously. All the fundamentals of *Synchronized Automata Theory* and *Statecharts* are cited. I develop an extension to these Theories using the hierarchical structures of *Trees* and model my systems with the *Automata Theory* tool. Finally, a formalization of *Automatically Extracted Signs* and *Rhythmic Phrases* serve to explain my solution to the main problem.

In Chapter five, all the implementation issues based on the theoretical tools defined in chapter four are developed. The specificities of implementation framework, low-level analysis, and rhythmic phrase detection and storage are underlined.

Chapter six gives a small and superficial description about the experiments that guided my research. The practical results were of fundamental importance to create systems that are effective in a real life 'on stage' situations, which was my primary objective. All the discussions about interaction is based on these experiences.

In the seventh Chapter, I present a detailed description of two case studies that prove the effectiveness of my approach. Each case study is a specific system, modeled by the tools developed in Chapter four and tested in real life videotaped situations. One of them has been accepted in three categories of presentation of SIGGRAPH 2009 ([Krakowski 09]). Two pieces produced at each case study are analyzed using a graphical representation and the practical issues regarding interaction are discussed.

Finally in the eighth Chapter, I summarize the results of this thesis.

Chapter 2

Related Works

In this chapter I present an overall view of some Computer Music works related to my research. My focus are the Interactive Music Systems (IMS) but some questions raised in the Musical Interface field deserve our attention as we will see. I also cite other related results to create a better portrait of the state-of-the-art.

We now concentrate on the Interactive Music System's field.

Some surveys were of great importance. [Roads 85] describes what now can be considered the origins of Computer Music. Besides the great difference in computational power, it is surprising how his words are still so actual. Talking about the use of "click track" in a tape performance he states: "A computer system that could use AI techniques to track tempo or listen for other musical cues could bring spontaneity and flexibility to the performance of computer music."

One of the motivations of this work was to scape from the click tracks, as I mentioned before.

The historical overview he presents cites the existence of Musical Automata that dates back to the second century BC. Other experiments worth noticing are the Norwisch programmable carillons of the thirteenth century; the mechanical spinet and drum set built by Leonardo Da Vinci; the Guido d'Arezzo look up table method; and the games composed by S. Pepys and W. A. Mozart.

Coincidentally (or maybe not) I used the Automata Theory to model the "modes of interaction", which, in simple words, can be considered as musical games. I will not go further into the resemblances between these historical cases and my research but I think it worth noticing them.

Roads classifies the areas of activity in four types:

1. the intelligent composer's assistant,
2. responsive instruments,
3. analysis and generative modeling of music,
4. recognition and understanding of musical sound.

In the second topic, which is of our interest, he cites the work of George Lewis (discussed further on) as being a pioneer in improvised computer music.

Later on, Roads goes further, and develops a comprehensive description that covers several aspects of Computer Music such as performance software and signal processing ([Roads 96]).

Two other important references on Interactive Music Systems are Robert Rowe's books [Rowe 93] and [Rowe 01]. The detailed description of Cypher, his musical system, and the musical examples he offers in these books were an important reference to the present work.

His approach to Rhythm using beat tracking, meter induction and quantization diverges from my point of view. Also, I will not be able to cite all the system descriptions there presented.

On the other side, there, we find the well accepted computer music systems' taxonomy:

1. Score-driven vs. Performance-driven systems.
2. Instrument vs. Player paradigms.
3. Transformative, Generative and Sequenced response methods.

I will use these axis, specially the first two, to classify the systems here presented. The ones I built during this research are all Performance-driven and use the Instrument paradigm, but the response method varies and I shall comment on it when necessary.

I start with a more detailed discussion about four Interactive Music Systems that are, for different reasons, more closely related to my approach. Then, I present articles that, although dealing with Musical Interface questions, have a strong relation to my point of view. Finally, I comment on other related issues. My aim is to use these works as reference points to locate my research and this is done gradually during this chapter.

2.1 Interactive Music Systems

The four works I chose to make a more detailed discussion about have one point in common: they work on stage. I expect the same from my results and this was the fundamental criteria that entirely guided my research.

All systems are described in some detail and analyzed using the following perspectives:

- their aesthetic motivation and flexibility;
- the rhythmical aspects of their musical responses, and;
- if they can, and how they deal with form issues.

One of the reasons why I chose these four systems is because their musical results can be found on the web or in CD recordings which are cited in my bibliography. My analysis is based both on the systems' scientific descriptions and on these results.

2.1.1 Voyager

George Lewis is an ubiquitous reference in the interactive music literature. Even in that primary survey above cited, [Roads 85], Lewis stands as the example of a pioneer musician-software engineer.

In [Lewis 00] he presents his interactive system, the *Voyager*. Its first version has been developed at the Studio for Elektro-Instrumentale Musiek (STEIM) between 1986 and 1988, and ameliorated with time.

In his words: "Musical computer programs, like any texts, are not 'objective' or 'universal', but instead, represent the particular ideas of their creators". In this case, his ideas are guided by, what he calls, *The Aesthetic of Multidominance*. He cites the artist and critic Robert L. Douglas as the formalizer of an African-American aesthetics that has its basis in the notion of "multidominant elements" which can be roughly defined as "the multiple use of colors in intense degree, or the multiple use of textures, design patterns or shapes"([Douglas 91]). This aesthetic principle is what rules the behavior of Voyager. In his words: "First, the *Voyager* program is conceived as a set of 64 asynchronously operating single-voice MIDI-controlled 'players', all generating music in real time."

I do agree it is important for a musical program designer to take into account what are the aesthetic values that guided his or her research. I let clear what are the aesthetic motivations of my work. On the other side, I also believe the computational results of one research shall be used in other aesthetically different contexts by other developers. That is why I described in details my results using mathematically formalized tools.

Unfortunately I did not find any detailed description on how the Voyager works computationally. In [Lewis 00], the author describes only two subroutines: *setphrasebehaviour* and *setresponse*. The first one, at each 5 to 7 seconds, resets the system behaviour, or in his terms chooses a new ensemble of "MIDI-players". In his words, this subroutine sets this new ensemble's "choices of timbre, the choice of one of 15 melody algorithms, the choices of volume range, microtonal transposition, tactus (or 'beat'), tempo, probability of playing a note [etc.]".

On the other side, the *setresponse* subroutine deals with the MIDI input signal (a pitch-to-MIDI device translates the musician's audio information to this protocol). Both the raw low-level MIDI information and "a mid-level smoothing routine that uses this raw data to construct averages of pitch, velocity, [...]" are used to build a response based on the input information. Unfortunately, the way this response is built and how the input affects the output is not detailed in this reference.

The rhythmic structure generated by the system is described as "moving in and out of metric synchronicity, with no necessary arithmetic correlation between the strongly discursive layers of multirhythm". As we can perceive by listening to the disc [Voyager 93], the musical flow generated by the computer do not follow a steady beat nor any kind of conventional metric standard. Instead, small pieces of structurally coherent groupings (timbre, rhythm, melody, or harmony based coherence) interleaves themselves during the musical discourse. This coincides with the implementation description given above.

In my case, this approach would not work since I need to have a steady beat response.

The form of a piece played with Voyager is also embedded in its implementation and is defined by that alternation mentioned above. The choice of doing so is based on the core of the Multidominance Aesthetics, in Lewis words: "It is to be noted that the work of many important African-American improvisers - in particular Cecil Taylor, John Coltrane and Albert Ayler - exhibit a notion of extended form that involves the sustained use, often for very long periods, of

extremely rapid, many-noted intensity structures." And clearly this guides his form approach.

One of the aims of this thesis is to allow the musician-improviser to decide what will be the form of a piece. This can be done previously in a compositional-driven approach, or on stage, during the presentation. In my case, even if a composition is outlined and guide the piece's structure, its representation exists only in the mind of the musician who has the responsibility of using and switching the interaction modes to create each part of the piece. Because of that, the form must not be embedded in the implementation as happens with the Voyager.

Nevertheless, we can notice a resemblance between the small alternated "algorithmic behaviours" of that system and my concept of modes of interaction discussed later on.

It is clear from his description that, following Rowe's taxonomy, this system uses a player paradigm. More than that, because of its autonomous generative capacity, the author supports the system is not hierarchically subordinate to the musician, which is in conformity with an important principle of the Multidominant Aesthetics. The author also makes a comparison between the authority-diffused interaction of musicians in a Gamelan ensemble and the authority-centered paradigm of a standard European orchestra identifying his system with the former example.

In my case, the questions about authority and subordination cannot be posed, since I use the Instrument paradigm and do not try to mimetize or expect a Player's behaviour.

Lewis claims that the communicative mode implemented in his work can make "the emotional state of the improviser [...] mirrored in the computer partner, even if the actual material played by the computer does not necessarily preserve pitch, duration or morphological structures found in the input". This means the concept of reflexivity, that I will comment again further on, is somehow a concern of this author. It is possible to find certain aspects of reflexivity in some of my results, as we will see.

Finally, I shall say I personally liked very much the recording [Voyager 93]. Despite my ignorance with respect to Contemporary Music, some sonic and interactive results he obtained seemed astonishing to me. I shall also admit, I did not like everything I heard in the disc but the overall impression was positive. Although I do not share the same aesthetic direction, it is important to notice the artistic result assumes a central role in both researches.

2.1.2 Cypher

In [Rowe 93], Robert Rowe presents his system, called *Cypher* and compares it with other Interactive Music Systems. The first thing we shall observe is that, while presenting the system, Rowe talks about making *Music* with it. In any place that I have noticed, he tries to characterize the type, genre, or aesthetic background of the music he wants to do with Cypher. One could think his goal was to reach an aesthetically flexible system that deals with, if not all, many kinds of music, but unfortunately this is not the case as we can perceive from his piece's descriptions and recordings.

His presentation of Cypher is extensive, profound and out of the scope of this research. I now highlight some important facts about it.

In his own classification, Cypher is performance-driven and uses the player paradigm. It

uses all the three synthesis methods: Transformative, Generative and Sequenced.

He divides the system in two components, the listener and the player. The former contains all the analysis section made over a stream of MIDI data (pitch-to-MIDI is used in real situations) with the use of several features. The player consists of an algorithm-based response to the result of that analysis. The connection between listener and player defines a *state* of the system.

To build whole pieces, Rowe describes two methods: score-orientation and improvisation.

The author is categoric when defining score-orientation is different from score-following. In his definition, the former consists on finding predefined marks in the input stream that make the system change to the *next* state, and the latter presupposes matching every event coming from the musician and generating a predefined sequenced response.

On the improvisation situation, the author reports two possible cases, one in which an *internal software critic* method controls the change of states and another where he acted as the operator making the state changes himself.

At this point a digression must be made: the above emphasized terms are important to my work. Clearly, Rowe describes Cypher as what I will call a multi-mode interactive system. Each of these states correspond, somehow, to my concept of mode of interaction presented later on. In the score-oriented case, Rowe is using what I will define as a sequential-mode interactive system. In the improvisation case, he reports an internal software critic mechanism which somehow can be linked to my concept of meta-mode of interaction in the sense that it controls the state changes.

These similarities encourage my belief that my approach is well fundamented. A quote about an improvised piece involving Cypher and the avant-garde saxophonist Steve Coleman substantiates even more this belief: "The most intriguing result of this performance and the rehearsal leading up to it was the speed with which the jazz players were able to pick up on the transformations tied to features of their playing and to trigger them at will. Connections between the Cypher listener and player were made onstage: a human operator (myself) performed state changes and other major modifications to program behavior." ([Rowe 93], p. 73).

Rowe reports the speed with which the jazz player understood how to control the answer from the machine and used it at his *will*. If this is a general tendency of the jazz players, what they might want is a system with clear musically-controlled interaction results that can be used when they choose so. The focus of this thesis is to show that a system with simple interaction modes and a musically-controlled method can be effective for musicians who deal with improvisation to build improvised music pieces. In my solution, I eliminate the figure of an external operator, as happens with the Cypher, and transfer its control function to the musician himself. In the situation above, not only the player would be able to trigger the answers, but also, he could choose what interaction scheme to use.

The way Rowe deals with rhythmic issues is not completely clear. In [Rowe 01] the author takes more care on aesthetically contextualizing each of the systems he presents. In the case of Rhythm, he uses the Western-music standard approach of beat-tracking, rhythm quantization and meter induction. On the other side, his insistence on not defining Cypher as score-driven indicates beat-tracking is not the most important location method used by him in Cypher.

In my approach I did try to use beat-tracking but it became clear that it was not the focus

I wanted to give. The experiences I had let clear that a beat-tracking problem can ruin a whole presentation, and that is why I chose the Instrument paradigm.

Unfortunately, I could not have access to more than just one piece played with Cypher ([Cypher 94]). Considering this and other pieces descriptions in [Rowe 93], p. 66, I imagine this system is used mainly in non-steady tempo Contemporary music contexts.

2.1.3 Continuator

For sure one of the characteristics that makes François Pachet's Continuator a powerful interactive music system is its capacity to learn style. By watching the system being used in a wide range of situations (from contemporary music and avant-garde jazz scenes to kindergarten experiences) we can grasp its adaptability.

One of the motivations for building this system was to develop an instrument that addresses explicitly this issue: "providing real time, efficient and enhanced means of generating *interesting* musical material". The author shows a clear will of achieving a *good* musical result, that can be used in real life. This will is shared by me when developing this research.

Pachet divides the musical system's domain into two categories: the interactive musical systems and the music imitation systems (term found in [Pachet 02] also called composition systems in [Pachet 02/2]).

In the former class he groups the systems that transform, in real time, "musical input into musical output" and that have as a common drawback the inexistence of memory. In other words, these systems do not use acquired information during the interaction with its users and, because of that, "the music [they generate] is strongly correlated with musical input, but not, or poorly, with a consistent and realistic musical style." (in [Pachet 02/2]).

I locate my systems in this class. I did try to include memory in the framework presented in this thesis, but the results were poor and I decided to focus on other issues. Nevertheless, I wish to go in this direction in further works.

On the other side, the imitation systems he describes are those capable of learning how to compose in a certain style. Its drawback, in his words, "is that they do not allow real musical interaction: they propose fully-fledged automata that may produce realistic music, but cannot be used as actual instruments."

Continuator "is an attempt to combine both worlds: design real-time interactive musical instruments that are able to produce stylistically consistent music."

The main idea behind it is to model, in real-time, the musical input using an Extended Markovian Model. The use of this method in composition systems is not new, but the possibility of using it to deal with information that comes from the musician during the interaction experience is what makes this system flexible to changes in rhythm, dynamics, harmony, or in more general and subjective terms, in style.

I give an overview on how the system works. The raw musical stream (represented using MIDI protocol) is segmented in phrases. No symbolic information is used, so the system do not use the canonical theory-based matching processes such as beat-tracking, meter induction, or transcription on these segments. Instead, the *reduction functions* such as pitch, duration,

velocity (or possibly the Cartesian product of them), are used to represent the input data in a set of tree structures.

Each branch of each tree represents a sub-phrase found in the input segments (for further details see [Pachet 02]). All the phrases are continuously parsed into these trees so all the interaction history is stored there. The root of these trees store a pointer to a continuation note, the note that continues the respective sub-phrase extracted from the input segments. In other words, the roots indicate all the possible continuations that were found in the musician's playing. This models a Markovian distribution for the probability of each continuation note. When a new phrase is received, the system checks if it matches (of partially matches) a branch of one of those trees and, based on each probability, chooses a continuation note using the root's information. The search for a next note can be done recursively. Discontinuities are handled by different strategies.

The system has also a "hearing" capacity modeled by a fitness function that calculates how much the next note "agrees" with it's context.

It is difficult to classify this system with respect to Rowe's taxonomy. For sure it is Performance-Driven, and uses Transformative and Generative methods. But does it use the Instrument or Player paradigm? By listening to the various recordings (video section of [Continuator] and audio [Continuator Audio]) we observe it mimetizes a Player behaviour (see Music Turing test on [Continuator]) although its creator define it as an Instrument.

Probably there is no answer for this question, but its dual character might have origin on the fact that the Continuator act as a "mirror", or in the words of its author, a Reflexive Interactive System. In this sense, one of its utilities is to share this abstract image the system builds from one player to interact with others, as is described in [Pachet 02].

The rhythmic aspects documented in [Pachet 02] have been ameliorated in a newer Continuator version but this have not been published yet. I describe them as they were before. Pachet divides the rhythmic behaviour of his system in four modes:

- Natural Rhythm: "The rhythm of the generated sequence is the rhythm as it was encountered during the learning phase [...]"
- Linear Rhythm: "This mode consists on generating streams of eight-notes [...]"
- Input Rhythm: The output rhythm is a copy of the rhythm of the last input
- Fixed Metrical structure: Predefined measure and tempo.

In my systems, I avoided predefined tempo or beat-tracking as in the second and fourth items. I do use the input rhythmic information to generate output like items one and three as we will see further on.

An important characteristics of the Continuator is that it does not deal with form issues. In fact, as its author states: "The responsibility for organizing the piece, deciding its structure, etc. are left to the musicians. The system only 'fills in the gaps'[...]". In this sense we will not expect it to be used alone to compositional applications.

One of my case studies presented here uses the Continuator as a mode of interaction. My wish is to be able to complement systems like this one by allowing the user to musically control (one or many of) them in an improvisation situation. I hope to be able to use again the Continuator in further works.

The recordings already cited, in my opinion, show how amazing are the musical results achieved with this system. As said before, the questions about form cannot be criticized, but the melodic solutions the system gives in the soloist mode and the harmonic paths it builds as accompanist are beautiful. Also, as an user in an interactive situation, I was very satisfied by the harmonic richness it brought to my experience.

2.1.4 Haile

Gil Weinberg and Scott Driscoll were the creators of Haile, a robot-percussionist ([Weinberg al. 05, Weinberg & Driscoll 06, Weinberg al. 06, Weinberg & Driscoll 07, Weinberg al. 07]).

Although it has also been used as a Marimba robotic-player, its first and more documented function is to act as a Middle Eastern Drum robotic-player.

In this context, Haile is designed to interact with Darbuka (Middle Eastern drum) players in an improvisational manner coherent to the aesthetics of the Middle Eastern percussion ensemble. In [Weinberg al. 06] the authors describe some specificities of this kind of interaction: "Most of the rhythms [played by this ensemble] are based on 4/4 meters [...]. Others are based on odd meters such as the Karsilama (9/8) Sha'bia (12/8) or Darge (6/8). The skills of the players in the ensembles are often judged by how well players manipulate the basic rhythm with variations through the use of dynamic, syncopation, accent, and the overlay of other rhythmic variations."

A typical Middle Eastern percussion composition might include a sequential exposition of different rhythms separated by bridges played in unison by all the musicians which are encouraged to add their variations to the previously rehearsed structure.

We shall notice this is the only system I found in literature that is also specifically designed to interact in real time with percussion instruments as is my case. The fruitful exchange of information with Scott was important in the early stages of my research.

The mechanical issues intrinsic to the robotic approach do not have any relation to the present work and I skip to the questions regarding the low-level analysis of percussion sounds.

In [Weinberg al. 06] this analysis is directed to the Darbuka instrument above mentioned, while in [Weinberg & Driscoll 07] and [Weinberg al. 05] it is designed to deal with the Native American Pow Wow drum, a low-pitched multi-player instrument.

The attack detection is done using the MAX/MSP *bonk~* object. In [Weinberg al. 05] they report that "Pre-filters and other [...] parameters were extensively tuned to optimise detection accuracy, which was complicated by loud hits masking softer later hits, and other non-hit noises from the hands rubbing the skin or additional ambient sounds." These were the reasons I did not use this object and developed our calibration approach detailed in chapter "Computational Issues".

After that, they also classify the input sound. In the case of the Pow Wow, the low pitch

stroke has a fundamental frequency of about 60Hz, and *bonk*∼'s frequency resolution (using a 256 point window) is not suited to deal with this information, so they developed a larger window (2048) FFT analysis.

The method I use when sound classification is needed do not follow this approach. Nevertheless, still in this chapter, we will comment on the classification of Pandeiro sounds in a broader scenario.

Similarity measure ([Tanguiane 93]) and stability ([Desain & Honing 02]) are extracted from the input and serve to guide the rhythmic generation.

Here a digression must be made. The similarity measures I use are described further on and are not based on the same reference.

On the other side, the idea of classifying the stability of a rhythmic pattern seems strange to us. In the very beginning of [Desain & Honing 02] the authors cite examples of rhythmic patterns that are "much easier to recognize and reproduce". Maybe because of my percussionist background, this sounds completely bizarre, and even more if said without any cultural contextualization. Further on, in the chapter "Rhythm" of this text, we will see why this contextualization must be done. There, I narrate an experience that happened with Tristan Jehan (and it was reported in his thesis [Jehan 05]) where some of his colleagues were unable to "understand" a Brazilian rhythm very "simple" to us.

Besides that, I could not find the place where the creators of Haile explain how these stability measures are used in the generation process.

The interaction between Haile and the other musicians is done using what they defined as the "modes of interaction" of the system. Their design is based on the aesthetic aspects found in the genre previously mentioned. I superficially describe the six modes found in [Weinberg al. 06]: *Imitation*, *Stochastic Transformation*, *Algorithm Morphing*, *Beat Detection*, *Synchronized Sequencing* and *Perceptual Accompaniment*.

The first four modes are sequential. In *Imitation* mode Haile repeats, after two seconds of silence, what has been played by the musician. In *Stochastic Transformation* the robot "stochastically divides, multiplies or skips beats in the input rhythm, creating variations that preserve the original feel of the beat [similarly to] the manner in which humans transform rhythms in the Middle Eastern percussion genre" (in [Weinberg al. 06]). In the *Algorhythm Morphing* mode, the system gets pre-recorded playings and morphs them together creating also style consistent phrasing. *Beat Detection* mode is commonly used in a "bridge" situation. It lets the musician inform the beat and tempo so that Haile can join the ensemble synchronized to the others.

The last two modes *Synchronized Sequencing* and *Perceptual Accompaniment* work in parallel with the input. In the first of them the robot plays just a Midi file, working as a sequencer. In the second, it listen to what it is being played and adapts itself. If the density of the input notes is high, the robot play sparser notes, and vice versa. The loudness is directly correspondent. Also the robot can play some call and response games during this mode.

We shall notice I adopt the same term "modes of interaction". In my case, I will describe it in details and give a very precise mathematical definition, as I believe this is a central concept in any interactive music research.

In my earlier experiences, I did implement also an Imitation mode but I did not use in more recent works. The Synchronized Sequencing technically resembles one of the modes used in my case studies but I do not deal with robotic generation, so aesthetically, the results are different.

We found two compositions where Haile plays the Pow Wow drum in the Internet site [Haile Web] which share, both, the same Middle Eastern compositional structure described above. The pieces where Haile plays the Marimba instead of the drum, will not be addressed here.

In the "Pow" composition described in [Weinberg al. 05] both the player and Haile play the same Pow Wow drum. The piece start with the Imitation mode followed by a call-and-response improvisation and ends with the musician and Haile taking turns as soloists. The video used as reference in the bibliography could not be found. I found the video [Haile Pow] entitled "Powwow". Because of its editing, it is difficult to understand the structure of the piece.

The Jam'aa piece is designed for Haile and two Darbuka players and is better documented in [Weinberg al. 06] but the videos I found did not match the entire description of the piece neither. The videos [Haile Jamaa1, Haile Jamaa2, Haile Jamaa3] show some extracts of it, but again, because of its editing, I cannot have an idea of the entire piece.

It is clear that, in both works, the modes of interaction appear successively in accordance with the Middle Eastern composition paradigm described before. Unfortunately, I did not find any description on how these systems change from one mode to the other and if that is a musician's choice or not.

2.2 Musical Interfaces

We shall, now, comment on some works about Musical Interfaces that were important to this thesis. Usually the term Interface is associated to a physical object that is used as a medium to carry information from the user into the computer. In this thesis I propose to use musical information to control it. More specifically, in my case, Rhythm is the Interface. This idea of using sound as an Interface can be found in [Di Scipio 03] although his approach differs from ours.

On analyzing the relationship between human users, gestural interfaces, and generative software, in [Wessel & Wright 00], David Wessel and Matthew Wright propose to see the whole interaction as an overall adaptive system. This means we shall not ignore human capacity of adapting him/herself to a given computer behaviour. During the experiences that directed my research we could observe the importance of this human adaptation, that, together with technical ameliorations, led me to use my systems in real live performances.

Another important concept they describe in this work is the *one gesture to one acoustic event* paradigm which rules the relationship between musicians and traditional acoustic instruments.

In computer music, this paradigm is broken and complex sonic responses can be produced by just one gesture. In my case, instead of gestures, I have sounds as the cause of interaction, but still, I face the same questions raised from the break of that paradigm. Projected video images can be useful to deal with some of those questions as we will see in one of my case studies. Other researchers already addressed this question ([Smalley 86]).

Another important reference in the field o Interfaces is the work of Sergi Jordà. He collaborated with Marcel-li Antunez to build the one-man-show multimedia piece *Afasia* inspired in Homer's Odyssey. In [Jorda 02], Jorda explains how the piece/system works: The performer uses an exoskeleton equipped with several sensors to control MIDI-controlled robotic instruments, a CD and a DVD player. The piece's structure is is divided in many *island-states* (based on the Odyssey) with different interaction behaviours. Each of theses states is formed by several *blocks* of MIDI tracks that can be played in a permutable order. A *control-track* defines how the user interacts with these blocks. The performer can interact with each "island" and can "navigate" between them in a main menu. Jorda states this navigation brings many "innovative solutions" but, unfortunately, he omits them in this paper.

We could trace a clear parallel between this system and my results. The island-states are related to my modes of interaction. The blocks controlled by the control-track give their interactivity identity. Unfortunately, I could not find the description about this interactive main menu but its function is clearly the same as of our meta-mode. The evident differences are that my systems are controlled through music instead of the exoskeleton, and that it is designed for an improvisation situation, and not as a narrative piece.

Other important works by Jorda worth noticing, such as the ReacTable ([ReacTable]), and several earlier results he summarizes in [Jorda 02/2]

As I was interested in practical results I had to deal with many questions regarding an 'on stage' situation. In this sense, I shall cite a paper by Perry Cook, [Cook 01], where, based in his 15 years of experience as an Interface designer, he states some "Principles for Designing Computer Music Controllers". I agree with him on some of these principles I now state:

1. Smart instruments are often not smart

2. Instant music, subtlety later

3. Wires are not that bad (compared to wireless)

While designing my modes of interaction, I realized the most complex ones, with many possible behaviours, just did not work, while the simple ones, with very clear response, were effective during interaction. In fact, this reinforces the first two principles. Trying to develop "smart" behaviours took us away from practice, while very simple schemes generated music right away, and from that I could complexify my implementation based on effective results.

The third principle is just a good tip: using wireless microphones when you need to extract information from its audio can let you in a very bad situation 'on stage'.

[Ryan 91] from STEIM also deal with Interface design issues. In [Paradiso 99] the author describes Tod Machover's Brain Opera built with several different interfaces. Another ensemble worth noticing is the [PLOrk 06] Orchestra, from Princeton.

The links between percussion and interfaces are many. Drum controllers such as MIDI drum pads can be found in the market. [Kapur et al. 02] developed the ETabla, a specific Tabla-like controller used for sound and image generation. In [Kapur et al. 04], they expand this approach to other Indian instruments.

[Aimi 07] builds hybrid instrument-controllers. Percussion instruments are damped and their sound is convolved to pre-recorded impulses as an strategy to scape from triggering-based techniques. A Frame Drum similar to a Pandeiro is one of the instruments used.

Many other references could be cited here. STEIM, cited above, and a whole conference (New Interfaces for Musical Expression) are devoted to this area of research, but this is out of the scope of this thesis.

2.3 Form

Here I make a small digression to compare the several approaches to Form that are presented in those described works.

The *Voyager* alternates that series of "behaviours" based on the Multi-Dominance aesthetics. The user do not have any choice in this alternation.

Using *Cypher* in a score-oriented situation means all the Form issues are predefined, on the other side, the Improvisation examples have Rowe as the Form manager.

The *Continuator* has some modes of interaction but there is no report about the use of more than one of them to build a musical piece.

The *Haile* changes from one mode to the other is not clear in its documentation, so as in the *Afasia* piece. What is clear is that these modes seem to be Sequential, this means the user might have the choice to go to the *next* mode, but not the choice of *switching to whatever mode whenever* he or she wants.

As we see, these Systems do not delegate the Formal choice of the piece to the performer. That is the motivation for building *Multi-Mode* Systems that can be *entirely* controlled by the improviser during the Musical Flow, through *Rhythmic Phrases*.

2.4 Other Related Works

In this section I present other related works from the Computer Music field. Both Accompaniment and Improvisational systems are briefly surveyed and compared to my approach. Then I present analysis works related to percussion. As part of that, I introduce the Pandeiro and comment on analysis issues specifically related to this instrument which assumes an important role in this research.

2.4.1 Accompaniment

As the name indicates, Accompaniment systems are designed to generate music that "supports", "accompanies" or "follows" the performance of a soloist. [Cabral 08] presents an in-depth comparative survey of several techniques used for Harmonic Accompaniment. There we can find the distinction between Arrangement systems, such as the [Band-in-a-box], the D'Accord guitar ([Cabral et al. 01]) or the automatic bassist [Geber 97]; and the Real-Time Accompaniment systems, also called Score-Followers.

This area has been inaugurated by the seminal works [Dannenberg 84] and [Vercoe 84]. The former develops a dynamical matching technique applied to keyboard input information while the latter uses digital signal processing together with sensor inputs installed in a flute in order to accomplish the accompaniment task.

Later, [Dannenberg 07] generalizes this technique to popular music which is "taken to mean music with a mostly steady tempo, some improvisational elements, and largely predetermined melodies, harmonies, and other parts." It is important to notice in the score-following approach, the system have both its own internal representation of the whole piece and a representation of the musician's score, and its function is to match them using the player's input music signal.

In the [Ircam] institute, this technique has been largely diffused.

We find, also, systems designed to create a percussion accompaniment. [Murray-Rust 03] builds his VirtuaLatin, an agent-based accompaniment system that uses a representational paradigm inspired in GTTM and is capable of adding Timbales tracks to a prerecorded salsa music.

In [Aucouturier & Pachet 05], the authors use concatenative synthesis ([Zils & Pachet 01]) allied to automatically extracted meta data representation to create a real-time drum accompaniment. The interactivity present in their approach, implemented through constraint-based techniques, is not found in typical score-following systems.

As we can see, these systems use the player paradigm in their designs, which differs from my instrument-driven approach.

Although some of the earlier works I developed have used a score-like paradigm to guide interaction, my main focus is on building systems that let the musician-improviser decide which musical path to trace during the performance. These musical paths are created by the choice of turning on and off the modes of interaction. In this sense, score-driven approaches are not suited for my task, nor the usual accompaniment paradigms.

Nevertheless, as I mentioned before (when discussing about the Continuator), it is possible to complement my approach by using modes of interaction that present interactive capability such as that found on the Ringomatic or in [Cabral 08]'s systems and I hope to able to develop that collaboration in further works.

2.4.2 Improvisation

In this thesis, I develop a framework to deal with improvisational situations. In chapter one, I let clear that, although the interaction between musician and computer can generate unpredicted results, which are welcome, the responsibility for building an "unpredicted", "surprising" and "creative" improvisational narrative is of the performer.

This means I do not expect Improvisation to arise from Interaction, and so I will not model the former to implement it in instantiations of the latter. My approach is to create simple modes of interaction, without no intention to mimetizing musicians behaviours, but just as building blocks. The reasons for doing so are two: First because building blocks must be simple. The time I would spend creating more complex ones would not let me have the enough quantity of them that is necessary to arrive to a Multi-Mode solution. Second because modelling Improvisation

means using a player paradigm which is not my case.

Nevertheless, I found important to list the several approaches for this modelling and the Interactive Systems that are based on them.

A first reference I shall notice is [Berliner 94] who is an endless source of jazz musician's quotes and is used as primary source to several researches.

[Walker 94] develops a model of Improvisation based on Conversation Analysis. This approach is based in the belief that "[i]n both improvisation and conversation, a participant must:"

- "Perform the distinct tasks of listening to other speakers, composing new utterances, and realizing the new utterances. The different tasks involved in spoken conversation (listening, composing, realizing) provide a functional decomposition of the task of musical improvisation."

([Walker 97], pp 1).

[Button 90] argues that an implementation of Conversation Analysis can create a "simulacrum of conversation" but do not capture the more subtle issues regarding real interaction. I also pose a criticism about this approach: the premise that a realizing phase is common to musical improvisation and conversation can lead to a strange result. In fact, we know music do not necessarily have "meaning" and so the idea of "realizing" a musical phrase can happen, but not in the same way a spoken phrase is realized.

[Pelz-Sherman 98] develops an analysis based on the concept of sender/receiver. He observes that "occasionally, performers seem to be sending and receiving equal amounts of information simultaneously to and from each other" (pp. xii). This corroborates with my criticism since in a conversation situation this cannot be observed.

In [Seddon 05], he reports several Modes of Communication used by jazz students during rehearsal and performance. This can serve as an inspiration for building a mechanism where the meta-modes are learned during interaction. I intend to do so in future works.

I make now a small summary of several Improvisational Systems. In [Wessel, Wright & Khan 98] and [Wright & Wessel 98] the authors describe a performance involving a Khyal singer and two computer musicians.

In [Thom 01], Belinda presents her *Band-out-of-the-Box* system based in a notational representation of the musical signal and the Theory of Believable Agents to "trade-fours" with jazz musicians. [Franklin 01] has the same objective with her *CHIME* system which is based in a neural network approach. Another system that deals with jazz situations is the Jam Session System for A Guitar Trio, [Hamanaka 01].

In [Murray-Rust 07], the author builds a complex model of Improvisation based on the Linguistic Theories of Pragmatics and Speech Act and applies it to build the *MAMA* system.

Finally, I shall observe that the work [Collins 06] was of fundamental importance to this thesis. The background built there served as a guide to the first steps on writing this thesis. In his approach, he develops an Agent-Based method applied to interaction. The use of a player paradigm differs from my approach.

Figure 2.1: Typical Pandeiro found at [Pandeiro.com].

2.4.3 Percussion

The Analysis of Percussive Sounds is a vast field ([Herrera al. 04], [Yoshii al. 03], [Sandvold 04], [FitzGerald et al. 02], [FitzGerald 04], [Gouyon, Pachet et al. 00], [Herrera et al. 03], [Zils, Pachet et al. 02]).

Some of the works are the devoted to percussion in general and others focus on specific instruments. That is the case of the Tabla which received much attention by the scientific community (eg. [Gillet et al. 03], [Chordia 05], [Chordia & Rae 08]).

On the other side, I did not find any written documentation about the History of the Pandeiro nor any scientific work about its sound analysis.

In this section I give an overview about this instrument and cite the sound analysis articles published by the staff of Sony/CSL Paris with my collaboration.

Pandeiro

The Pandeiro is a Brazilian percussion instrument which is part of a larger family of Frame Drums. These drums are popular in the Arabic world and can be found in many Mediterranean countries so as in Portugal. Probably it has been brought from the Iberic Peninsula to Brazil during the Colonial Period.

This instrument gain its identity as a Brazilian national symbol in the 1930's when the Samba genre was widely diffused in Brazil ([Vianna 95]).

Usually the Pandeiro is handmade, built with a skin over a cylindrical wood body of small height (about 4 cm) and small cymbals (also called jingles or platinelas in the original) equally distributed around this body (see Figure 2.1).

As happens to many percussion instruments, there is no universally diffused method for playing it. In fifteen years of work, I developed a didactic method to teach my technique where I divide the possible sounds of this instrument in six classes (see [Roy, Pachet & Krakowski 07/2]):

1. tung: Bass sound, also known as open sound;
2. ting: Higher pitched bass sound, also open;
3. pa: A medium sound produced by hitting the Pandeiro head in the center;

4. PA (or big pa): A slap sound, close to the Conga slap;

5. tchi: The jingle sound; and

6. tr: A tremolo of jingle sounds.

In fact, these classes can be grouped in three more fundamental ones, the 'tung', the 'pa' and the 'tchi' sounds from which the other three sounds (respectively 'ting', 'PA' and 'tr') are a special variation. I use this more simple grouping throughout this work.

I describe how to produce them:

1. The 'tchi' sound is produced by hitting the edge of the instrument. It is a high pitched sound (around 2 to 15 KHz).

2. The 'tung' is a low pitched (around 50 to 200 Hz) sound produced by hitting the skin half way from the center to the edge using the thumb or the fingertips. It can be compared to the bass drum sound.

3. The last sound is called 'pa' and it is produced by hitting exactly the center of the pandeiro either with the thumb or fingertips. It is medium pitched (around 400 - 1000 Hz) and is close to the snare drum sound.

One shall notice as the platinelas are attached to the body of the drum, they can always be heard in whatever stroke produced. This means there is a 'tchi' "attached" to all the other sounds.

A profound work for automatically classifying the six types of sounds has been made by the Sony/CSL-Paris staff with my collaboration ([Roy, Pachet & Krakowski 07/1], [Roy, Pachet & Krakowski 07/2],[Roy, Pachet & Krakowski 07/3]). In these works so as in [Cabral et al. 07] the general *bag-of-frames* strategy is improved using the EDS system ([Pachet & Zils 04], [Pachet & Roy 07]). Using this technique it was possible to solve the classifying problem even using just three features chosen by EDS.

Although this result have been implemented in real time, I chose not to use it in interactive tasks for one reason: the problem of rhythmically-controlling the system was already enoughly complex just by using attack detection. The improvements of 6-sound classification are left for future investigation.

Nevertheless, in Chapter five I use an ad-hoc 'pa'/'tung' classification in the same spirit of the distinction snare/bass-drum found in the literature ([Zils, Pachet et al. 02]). Again, this was just an effective solution to be used in interaction experiments since the 6-sound classification was discarded.

Chapter 3

Rhythm

This Chapter is dedicated to a more detailed investigation on several works that have Rhythm as its central focus. I try to present different perspectives that can assist on building Interactive Systems. In the end of the Chapter, I present the point of view that best suited the aesthetic context I am immersed in.

The results of this discussion are not used directly used in my research, but serve as a very important background since I focus on building Rhythm-Based Interactive Systems.

I start the Chapter giving an overview about Music Perception and Cognition. Then, I present some definitions of Rhythm and finally, I investigate four different approaches about the subject.

3.1 Music Perception and Cognition

Listening to music seems to be an easy task for a human being. Understanding what are the perceptual and cognitive processes behind it has shown to be a challenge for researchers in the domain of Psychology of Music. This comprehension is fundamental for many other fields such as Computer Music.

The issues addressed in it range from Psychoacoustics to Harmonic Cognition, from Behavioral to Gestaltian approaches, from motory to emotional facets of music listening.

In my case, I will present only some topics that are related to temporal aspects of music perception and cognition. First of all we shall understand the very basic notions on how human apprehend time.

Psychological Present In 1890 William James published a seminal work in this field. He cited the work of a certain E.R. Clay who had coined the expression "specious present". In fact Clay was a pseudonym of the amateur philosopher Robert Kelly who, together with Shadworth Hodgson, influenced the work of W. James ([Andersen 09] gives an overview of the historical aspects of this issue).

The term is defined by James as "the time duration wherein one's perceptions are considered to be in the present" and is also known as the "psychological present".

The very first definition of this term by E.R. Clay uses musical notions to describe it: "All the notes of a bar of a song seem to the listener to be contained in the (specious) present. [...] At the instant of the termination of such series, no part of the time measured by them seems to be a past."

These researchers propose that the notion of present conceived by humans has a length in time and could be considered as a recent past, which is opposed to the philosophical concept of Present as an instant without measure.

Alf Gabrielsson [Gabrielsson 93] explains the concept in a worth noticing way: " Clap any rhythm pattern you like. Then make it successively slower and slower, and it will not take long until you discover that it gets very difficult to clap it any longer; the pattern dissolves, leaving only a number of isolated events." If we agree on this theory, the reason for this dissolution is that the pattern's length, when the temporal stretching reaches a certain point, violates the limits of the length of the psychological present.

This limit was one of the main subjects studied by the psychologist Paul Fraisse. In [Clarke 99] we find a overview of his work. From there I quote "Fraisse drew a primary distinction between the perception of time and the estimation of time. The former is confined to temporal phenomena extending to no more than about 5 sec or so, whereas the latter relies primarily on the reconstruction of temporal estimates from information stored in memory."

Based on Fraisse's ideas, Clarke proposes a division between *rhythm* and *form*. In Jazz, Choro, Samba, Western classical music and many other styles, the concept of form emerges as a way of sectioning a piece in smaller parts, typically of one minute length in the case of popular music.

In fact, although [Cooper & Meyer 60] are talking about Western classical music, this could be extended to popular music as well: "Most of the music with which we shall be concerned is architectonic in its organisation. That is, just as letters are combined into words, words into sentences, sentences into paragraphs, and so on, so in music individual notes become grouped into motives, motives into phrases, phrases into periods, etc."

This architectonic characteristics of music has a counterpart in mathematics, the concept of multi-resolution. In fact, [Smith 99] develops a representation of musical rhythm based on a time-frequency approach that leads to the use of Morlet wavelets for this purpose. This approach worth noticing because links an important mathematical technique to a non-rigorous music architecture concept, which in my point of view strengthens both approaches. Despite that, I will not use this representation in the present work.

On the other hand, the division between rhythm and form will be useful for us, although no rigorous definition of duration boundaries between them will be adopted. Qualitatively, rhythm aspects will deal with small amounts of time, in the range of psychological present, and form aspects will deal with sections of music, keeping in mind they are part of the same structure. Still inside this structure, I could go even further on "microscopic" aspects of time, such as timing, timbre analysis, and pitch phenomena. I will address some of these questions, but they will not be a central issue of this thesis.

3.2 Definitions of Rhythm

Everyone that is engaged in the activity of making music, be it in a performance or in a compositional situation, using standard instruments or the computer, had to face the question of where in time will be placed the *next* note. In my point of view, this question has been continuously underestimated.

Defining Music is a difficult task. Informally, though, we could say music is about placing sounds in time. Many voices from various fields seem to agree on it. [Gabrielsson 93] says: "Music takes place in time, and temporal organization of the sound events achieved through rhythm simply *must* be of fundamental importance. In fact there is no melody without rhythm; melody thus presupposes rhythm." (emphasis in original).

[Cooper & Meyer 60] are even more categoric in the very first sentence of their book: "Every musician, whether composer, performer, or theorist will agree that 'In the beginning was rhythm'. ".

Another categorical phrase from one of the most important researchers in the field of Music Psychology can be read in [Fraisse 82]: "a precise, generally accepted definition of rhythm does not exist" so I will not try to find one.

Although it is impossible to be exhaustive, I will investigate some approaches found in various fields in an attempt to give an overall portrait of what is meant by *music rhythm*. Some surveys were of extreme importance, both [Clarke 99] and [Krumhansl 00] in a broader scenario, and [Pfleiderer 03] specifically focused in popular music.

First of all I define some categories of approaches. Although many of them are linked and interleaved, we could separate them in four groups:

- Theoretical: The standard top-down approach developed to deal with questions about notated Western tonal concert music of the 17th to 19th-century period.
- Experimental: Using laboratory experiments, try to affirm or deny Music Psychology cognition models based on the idea that the results of these experiments can be generalized to deal with real-life situations.
- Computational: Is focused on solving practical questions about retrieving, analyzing and representing rhythm from the digitalized audio signal applying these results to computer music systems and to the discussion of Music Psychology cognition hypothesis.
- Ethnomusicological: Includes cultural issues in the analysis and modeling of rhythmic cognition taking into account the cultural context these phenomena are immersed in.

To begin the discussion I summarize some definitions of Rhythm:

- Theoretical:

[Cooper & Meyer 60] "Rhythm may be defined as the way in which one or more unaccented beats are grouped in relation to an accented one." And somewhere else they explain it in other words: "To experience rhythm is to group separate sounds into structured patterns. Such grouping is the result of the interaction among the various aspects of the materials of music: pitch, intensity, timbre, texture and harmony - as well as duration".

[Lerdhal & Jackendoff 83] : They do not give a one sentence definition, but in the very beginning of the chapter "Introduction to Rhythmic Structure" they postulate "The first rhythmic distinction that must be made is between grouping and meter."

- Experimental:

[Parncutt 94] "A musical rhythm is an acoustic sequence evoking a sensation of pulse"

[Gabrielsson 93] "[A musical or auditory rhythm is] a response to music or sound sequences of certain characteristics, comprising experimental as well as behavioral aspects."

- Computational:

[Scheirer 98] Uses the words of [Handel 89] "The experience of rhythm involves movement, regularity, grouping, and yet accentuation and differentiation" (p. 384)

[Gouyon 05] "Rhythm is about recurring musical events occurring roughly periodically".

- Ethnomusicological:

[Iyer 02] Does not give a one sentence definition, but his proposition of seeing rhythm as an embodied cognition phenomenon is clear: "the act of listening to rhythmic music involves the same mental processes that generate bodily motion."

[Agawu 03] Also, does not try to theorize about rhythm in general, he defines the concept of *topos* which will be important to us: "a short, distinct and often memorable rhythm figure of modest duration (about a metric length or a single cycle), usually played by the bell or high-pitched instrument in the ensemble, and serves as a point of temporal reference" (p. 73).

I go further on the investigation of each approach given here.

3.3 Theoretical approach

Cooper and Meyer First we find the ubiquitous reference on rhythm, the work of Grovesnor Cooper and Leonard Meyer. They had an important role on formalizing the analysis of rhythm in a text book, whose purpose is to teach music, instead of just create a musicological theory. In fact the authors complain that "[t]here are many textbooks on harmony and counterpoint but none on rhythm"

I will give a quick overview on how they define the tools that will be used in their analysis. Firstly, the definition of Architectonic characteristics of music was above mentioned. Then they define:

- Pulse: "is one of a series of regularly recurring, precisely equivalent stimuli. Like the ticks of a metronome or a watch, pulses mark off equal units in the temporal continuum. Though generally established and supported by objective stimuli (sounds), the sense of pulse may exist subjectively. A sense of regular pulses, once established, tends to be continued in the mind and musculature of the listener, even though the sound has stopped."

- Meter: "is the measurement of the number of pulses between more or less regularly recurring accents. Therefore, in order for a meter to exist, some of the pulses in a series must be accented - marked for consciousness - relative to others."

- Beat: "When pulses are thus counted within a metric context, they are referred to as beats. Beats which are accented are called 'strong'; those which are unaccented are called 'weak'."

- Accent: "Though the concept of accent is obviously of central importance in the theory and analysis of rhythm, an ultimate definition in terms of psychological causes does not seem possible with our knowledge. That is, one cannot at present state unequivocally what makes one tone seem accented and another not." And as it is a central concept to this work, they add: "In short, since accent appears to be a product of a number of variables whose interaction is not precisely known, it must for our purposes remain a basic, axiomatic concept which is understandable as an experience but undefined in terms of causes."

- Rhythm's definition has been given before. It is important to notice they use *prosody*'s concepts as the reference to build the categories of rhythmic grouping (more specifically, some of the poetic feet). They also talk about the relation between rhythm and meter that appears to be based on accents: "[the] accents and unaccents, when they occur with some regularity, would seem to specify the meter. In this sense the elements which produce rhythm produce meter."

- Grouping has an important role: "Since this book is in fact concerned throughout with grouping - for that is what rhythm is - the following discussion of the general principles of grouping will be minimal." Finally they also state that: "Grouping on all architectonic

levels is a product of similarity and difference, proximity and separation of the sounds perceived by the senses and organizes by the mind."

Far from trying to postulate principles, the authors stress the fact that "analysis [is] an art rather than a science", depends on the skills of the analyzer, and define many aspects of performance interpretation which seems to be the final goal of this work.

There is one aspect on their work that deserves to be noticed: the word *rhythm* for them is strictly linked to the idea of *grouping* and not to the idea of *regularity*, *periodicity* nor *recurrent pattern*. Instead, the word *pulse* stands for non expressive repetitive stimuli, that when are regularly accented become *beat* and generate *meter*.

Lerdahl and Jackendoff I now concentrate on the work of Lerdahl and Jackendoff. Clarke presents it in these terms: "Quite apart from its importance in other respects, a significant contribution of Lerdahl and Jackendoff's *A Generative Theory of Tonal Music* [...] was its clarification of the elements of rhythmic structure in music, in particular the distinction between grouping and meter." It seems to be a consensus that the rhythmic aspects of their theory were the most valuable part of it.

Their whole work is based on the Generative linguistics theory " [which] is an attempt to characterize what a human being knows when he knows how to speak a language, enabling him to understand and create an indefinitely large number of sentences, most of which he has never heard before."

Using the same approach of this theory to music, they develop rules, analog to a *grammar*, through which music should be regulated in order to be well-formed. They claim that one of the positive aspects of their theory is that besides the well-formedness rules, they create preference rules that stand to modeling aesthetic (sometimes paradoxical) aspects of musical analysis.

As we have already noticed, the concept of *rhythm* for Lerdahl and Jackendoff can be divided in *grouping* and *meter*. As before, I will quote the definitions used to deal with these concepts:

- Grouping: "From a psychological point of view, grouping of a musical surface is an auditory analog of the partitioning of the visual field into objects, parts of objects, and parts of parts of objects. [...] Moreover, the rules for grouping seem to be idiom-independent - that is, a listener needs to know relatively little about a musical idiom in order to assign grouping structure to pieces in that idiom." They also state that: "Grouping structure is hierarchical in a non overlapping fashion (with the one exception mentioned above), it is recursive, and each group must be composed of contiguous elements."

- They define three types of accent:

  - Phenomenal accent: "[an] event at the musical surface that gives emphasis or stress to a moment in the musical flow."
  - Structural accent: "an accent caused by the melodic/harmonic points of gravity in a phrase or section - especially by the cadence, the goal of tonal motion."
  - Metrical accent: "[a] beat that is relatively strong in its metrical context."

And relate phenomenal to metrical accent: "Phenomenal accent functions as a perceptual input to metrical accent - that is, the moments of musical stress in the raw signal serve as 'cues' from which the listener attempts to extrapolate a *regular pattern* of metrical accents. If there is little regularity to these cues, or if they conflict, the sense of metrical accent becomes attenuated or ambiguous. If on the other hand the cues are regular and mutually supporting, the sense of metrical accent becomes definite and multi-leveled. Once a clear metrical pattern has been established, the listener renounces to it only in the face of strongly contradicting evidence. Syncopation takes place where cues are strongly contradictory yet not strong enough, or regular enough to override the inferred pattern."(my emphasis).

- Beats are "[t]he elements that make up a metrical pattern [...] It must be emphasized at the outset that beats, as such, do not have durations. [...] But of course, beats occur in time; therefore an interval of time - duration - takes place between successive beats. For such intervals we use the term *time-span*." It is important to notice that the idea of beat, differently from [Cooper & Meyer 60] is that of an element of a multi-resolution metrical grid. Each level of this grid would have equally spaced beats and the time-span from two consecutive beats in a certain level n should correspond to the time-span of two or three consecutive beats in the lower level $n-1$.

- Tactus is seen as a special level of this grid, and is analogue to the concept of pulse in [Cooper & Meyer 60]: "However, not all these levels of metrical structure are heard as equally prominent. The listener tends to focus primarily on one (or two) intermediate level(s) in which the beats pass by at a moderate rate. This is the level at which the conductor waves his baton, the listener taps his foot, and the dancer completes a shift in weight [...]. Adapting the Renaissance term, we call such level the *tactus*."

- Metrical structure is defined as "the regular, hierarchical pattern of beats to which the listener relates musical events." The function of meter is given: "The term *meter*, after all, implies measuring - and it is difficult to measure something without a fixed interval or distance of measurement. Meter provides the means of such measurement for music; its function is to mark off the musical flow, insofar as possible, into equal time-spans. In short, a metrical structure is inherently periodic. We therefore assert, as a first approximation, that beats must be equally spaced."

I shall make a digression to explain why beats cannot be equally spaced and why they refine this proposition. The problem of defining that the beats must be equally spaced is that it does not allow, for example, the existence of triolets in a 4/4 time signature. In fact, to deal with the problem of temporary subdivisions of the tactus, they define that below the tactus, the subdivisions does not need to be equally spaced during the whole piece. They cite the case of Brahms Clarinet Sonata (op.120, no.2, measures 9-11) where there are a series of triolets and a quintuplet. As they say quintuplets "are so rare in the idiom we are considering", they define it as an "extrametrical event", such as "grace notes, trills, turns, and the like." So they postulate

that these subdivisions below the tactus must be in two or three parts and need not to be extend all over the piece. Finally, in the discussion about the tactus level, they state: "Metrical structure is a relatively local phenomenon."

As we see from this exposition of concepts, Lerdhal & Jackendoff divide the concept of rhythm in two: *grouping*, *meter*. Only the latter phenomenon is related to *regularity*. In fact, *beats* generate the *metrical structure* from which the *tactus* is a special level.

I shall observe that Cooper & Meyer's work is also based on a linguistic construction. The difference is that prosody seems to deal with sonic questions such as intonation, and syllable rhythm. Instead, generative linguistics deals only with the symbolic representation of a language.

Recurrent Patterns My interest is on working with Rhythm, but the way those authors deal with it do not seem to have a direct connection with the idea of recurrent pattern, that we now investigate.

We must notice, in both works, Gestalt laws are applied only to explain the phenomenon of grouping. In fact, Lerdhal & Jackendoff explicitly use the "visual partitioning" analogy, while Cooper & Meyer talk about "similarity and difference, proximity and separation of the sounds perceived" in another clear analogy.

On the other side, the idea of recurrent pattern is always linked to the concept of meter, but always in an "indirect" or "induced" manner. In fact, Lerdhal & Jackendoff state that "Metrical accents" are induced by "Phenomenal accents": "the moments of musical stress in the raw signal serve as 'cues' from which the listener attempts to extrapolate a *regular pattern* of metrical accents". Cooper & Meyer state "[the] accents and unaccents, when they occur with some regularity, would seem to specify the meter."

These authors do not analyze the phenomenon of a recurrent pattern through the Gestaltian perspective.

I believe this has an explanation: the cultural background used for the base of their theory.

In a Brazilian cultural background, the examples of explicit recurrent patterns can be found in almost all the genres of "popular music". In some of these genres, our explicit recurrent patterns are also blended with European-like examples such as those described above. That happens, for example, in a Choro piece where the European meter is blended with a repetitive Pandeiro pattern that also defines the *groove* of the piece (see [Iyer 02] for the idea of groove as a localization phenomenon).

Although a complete investigation of this issue is out of the scope of this thesis, I advocate that the phenomenon of recurrent patterns should be analyzed through the optics of the Gestaltian laws.

Independently of being explicit or not, a recurring pattern in a musical situation seems to create the same relationship with the melody as in the Figure-Background relationship ruled by the law of Symmetry.

In this direction, we shall notice the work of Parncutt, that I describe later on. He defines the concept of *periodic grouping* which contrasts with that of *serial grouping*, related to the melody.

In the end of this Chapter I will put together some results that generate a unified model of rhythmic patterns, including the traditional meter structure here presented.

Criticism Here I present some criticism to the Theoretical Approach.

First we must notice they deal with notated music, not with music. In fact, many times we can see the term *musical surface* in [Lerdhal & Jackendoff 83]'s text as being directly inferred from the respective score, which does not happen in general.

I will not cite the many works that agree on this point of view. [Clarke 99] talking about the grammar rules of that work state that "Lerdahl and Jackendoff offer no empirical evidence for the operation of these rules, relying on their own musical intuition to guide them."

[Iversen 08] also proved that the grouping is not "idiom-independent" as proposed Lerdhal & Jackendoff.

[London 04] proposes the concept of non-isochronous pulse which serves to model several rhythmic structure such as the odd meters of contemporary jazz and macedonian music. This goes against the common assumption those authors use that a pulse must be isochronous.

Finally, [Iyer 02] makes a very strong critics in the very beginning of his work is: "A great majority of the research in music perception and cognition has focused on a rather narrow segment of human phenomena, namely the tonal concert music of pre-20th-century Western Europe, as filtered through contemporary European-derived performance practices. Hence we have an abundance of tonal-music-inspired models and representations for the perceptual and cognitive phenomena, focusing mostly on pitch organization in the large-scale time domain. Some well-known examples are theories of recursive formal hierarchies (Lerdahl & Jackendoff) [...]. Such models suppose that the cognition of music consists of the logical parsing of recursive tree structure to reveal greater and greater levels of hierarchical organization. However, because so much musical behavior in non linguistic in nature, music tends to challenge dominant linguistic paradigms, which reduce all cognition to rational thought processes such as problem solving, deductive reasoning, and inference. With its emotional and associative qualities and its connection to dance and ritual, music seems to provide a counterexample to such theories of mind.

While quite far-reaching in the case of Western tonal music, linguistics-derived musical grammars do not apply well to the vast majority of other genres of music. This non translatability is quite glaring in the cases of African-American forms such as jazz, rumba, funk, and hip-hop. In these cases, certain salient musical features, notably the concept of groove, seem to have no analogue in rational language."

3.4 Experimental Approach

Differently from the Theoretical approach, this one aims to sustain or refuted cognitive models using empirical data. As the methodical analysis of cognition issues is a hard task, the models to be tested usually focus on very specific characteristics of musical cognition, what makes the literature in this area assume a fragmented aspect.

Some synthesis of these analysis were successful and allowed a step forward in the general comprehension about rhythm. The work of Paul Fraisse is one of those cases and seems to be a primary reference on the field ([Clarke 99]).

I will present some results from this author as they appear in [Clarke 99]. Fraisse makes the distinction between *time perception*, that deals with phenomena about 5 seconds long; and the *time estimation*, that deals with longer phenomena which are linked to the reconstruction of information stored in the memory. Besides that he observed the importance of the body in the rhythmic cognition, tendency on categorizing a sequence of rhythmic intervals as being related by the proportion 2:1 and a series of other important phenomena. Finally I shall underline the study about the 600 milliseconds value, called the "indifference value", a threshold that qualitatively separates the set of temporal segments and that would be the natural choice when a subject is asked to tap a continuous pulse.

I shall also cite the work of Richard Parncutt ([Parncutt 94]) that methodically analyzes the issue of finding a pulse in an isochronic sequence of stimuli. Once again the 700-millisecond value is found to have special properties (which reinforces Fraisse's results).

In one of his experiments, Parncutt wonders about an unexpected interpretation many subjects made when asked to find where the metric accent should be placed in a pattern the author named "march" rhythm. [Pfleiderer 03] argues that this "misuderstanding" probably comes from the fact that this rhythm is very close to a shifted-phase version of a standard Rock'n'Roll pattern, and this cultural aspect is ignored in Parncutt's interpretation.

In this sense, Pfleiderer is questioning how far could we go into the search for universal principles of musical cognition without considering the subject's cultural context. He, then, cites the pioneer work of Alf Gabrielsson who includes popular music recordings in his experiments.

In [Gabrielsson 93], he surveys some of his results about Rhythm. His motivation is to study musical experience beyond what he calls *structural aspects*, those related strictly to musical notation. [Cooper & Meyer 60] is cited by him as an example of the structural approach.

Gabrielsson believes in the premise that musical experience is multi-dimensional. In the search for how many and which are the important dimensions that characterize this experience, he uses multivariate analysis techniques. In this survey he reports his experiences induced the representation of a rhythmic cycle as a point in a 15-dimension space. These cycles are generated not only through synthetic sounds but also played by a percussionist or extracted from popular music recorded pieces.

He sustains each of these dimensions can be seen as adjectives that characterize these cycles and they can be easily grouped in three types: structural, motional and emotional. The structural adjectives are linked to notational issues already mentioned; the motional ones are related to embodied responses generated by these cycles, e.g. walking, dancing, swinged, etc; the emotional adjectives are solene, calm, rigid, vital, joyful etc.

With his work, Gabrielsson proofs it is possible to deal more profoundly with rhythm cognition models by including culturally dependent information.

3.5 Computacional Approach

Many of the computational works that deal with rhythm have as their final goals practical results such as the automatic cataloging of musical genre, high level information retrieval or the development of interactive systems.

Some of them are related to the above mentioned approaches whether sustaining or denying those models.

I present some examples of works dedicated to rhythmic issues that will be useful when presenting my framework to deal with percussive monophonic signals.

First, we have the important paper [Scheirer 98] extensively diffused in the literature about rhythm. Scheirer presents a method to pulse detection in a polyphonic signal.

In this work, the author demonstrate his interest on including several musical genres, which seems to be ubiquitous in the computational approach, contrasting with the theoretical and the majority of empirical approaches.

Scheirer starts from the premise, which became largely accepted, that the rhythmic content of a polyphonic signal can be integrally represented by the amplitude envelope of each of its frequency bands. He passes the signal through band pass filters (six, in his case) covering the whole auditive spectra and calculates the derivative of the envelope of the amplitude of each of these filtered versions of the original signal. This derivative feeds a comb filter bank that serve as a series of resonators. The period of the comb filter whose output signal is the strongest one becomes a good candidate to the period of the searched pulse.

The author affirms the signals treated in this work are more complex than those analyzed in other works of that time, and so, the comparison between algorithms cannot be done. When reporting about the ineffectiveness of his algorithm on detecting the pulse of some "up-tempo jazz" examples, Scheirer observes humans are able to "induce" the pulse of complex signals even though there are not explicit periodic accentuation.

In this sense, Tristan Jehan, using the same algorithm in [Jehan 05], had problems on defining the pulse of a polyphonic maracatu music signal (a genre from the northeast of Brazil). By asking his colleagues to do the same task, he realized the misunderstanding was not restricted to the computer analysis.

This proves the phenomenon of pulse is not objective and the subjective aspects determined by the culture must be included in the analysis as he does in [Jehan 05].

Another important work focusing the rhythm is [Paulus & Klapuri 02] where Paulus & Klapuri present a similarity measure between rhythmic patterns.

First, they develop a way to segment a rhythmic pattern by calculating the periodicity of [Scheirer 98]'s amplitude envelopes without using the comb filter bank. The signal representation is done using three characteristic functions: the energy, whose perceptual correspondent is the signal's volume; the spectral centroid which corresponds to the brightness of the sound; and some MFCC coefficients.

Two rhythmic patterns are segmented and represented in the above mentioned way and the two resulting arrays are compared using a temporal warping technique.

This method has been tested successfully on electronic drum phrases played by an amateur

musician and on excerpts of real music. The authors arrived to the conclusion the spectral centroid is the best representation when dealing with rhythm, which agrees with Scheirer's premise.

The concept of musical recurrence is very important to Music. Foote & Cooper proposed in [Foote & Cooper 01] an important method to visualize the self-similarities of a polyphonic audio signal.

The authors partition the signal in equally long segments and represent each of them through spectral analysis (they cite Fourier or MFCC coefficients). They build what is called the self-similarity matrix by comparing the representative value of each segment to the value of all of the other segments.

They obtain a powerful visualization where we can easily see repetitions of rhythm and form structures, which reinforces the multi-resolution characteristics of the musical signal ([Smith 99]). As an application of this method, the authors show it is possible to find the tactus of some classical music and jazz excerpts.

[Gouyon 05] presents an important work in the field of Rhythmic Descriptors where he uses Periodicity Functions calculated over the audio signal to obtain tempo and meter information. Several works are also compared in his text. [Termens 04] also deals with the same area of research.

[Bilmes 93] develops an important work on Musical Expressivity. He models the phenomenon of Expressive Timing, implement it and applies this system to the analysis of the Munequitos de Matanza Rumba group.

[Pachet 00/2] applies an evolutionary approach to model musical Rhythm.

In the field of Information Retrieval, I can cite [Kapur et al. 05] and [Nakano et al. 04]. [Tsunoo et al. 09] and [Dixon et al. 04] are interested on Rhythm-Based Genre Classification.

Finally, although I do not agree on classifying rhythmic patterns by their complexity or stability, we shall notice there are several works on the literature that propose this approach: [Desain & Honing 02], [Toussaint 02], [Toussaint 04] and [Shmulevich et al. 01].

3.6 Etnomusicological Approach

Presenting this approach in a Computer Music thesis can sound unusual. Those interested in ethnomusicology generally will not reach computational results nor universalizing theories as happens in the previous approaches. But their view about rhythm includes a very important class of music: that based on rhythm ostinatos. Of course some of the above mentioned approaches include jazz, rock, rumba, etc, but few of them develop specific tools to deal with rhythm recurrence which is to us of central importance.

Firstly, I shall comment the formalization made by Kofi Agawu in the book "Representing African Music" ([Agawu 03]). One of his goals is to destroy the myth that African Music is interesting only because of its rhythmic characteristics. In this sense, he shows that, until then, the analysis of African genres always involved some kind of special exotic notation that did not allow the comparison of these genres with those from other ethnic origins. Because of that, he presents an analysis of West African dance/music styles using the usual music notation.

An important fact is that this analysis is always based on the dance-music pair and all the music considerations take into account issues related to the body of the dancer. The concept of *topos* developed by Agawu will be of extreme importance in this work.

In his words: "A *topos* is a short, distinct and often memorable rhythm figure of modest duration (about a metric length or a single cycle), usually played by the bell or high-pitched instrument in the ensemble, and serves as a point of temporal reference". As he underlines and as the Latin word itself induce, the *topos* is the temporal reference to musicians and dancers. It is not just in African styles that we find this element. The Cuban *rumba clave*, and the *surdo* phrasing in the Brazilian *escola de samba* are examples of *topoi* in their respective contexts.

Synchronizing musicians and dancers is a fundamental role rhythm ostinatos play in these cultures, but we can go further. Let's consider the *topos* can be implicit, not played by any musician, and just "induced" as Scheirer suggests. For example, the counting "ONE, two, three, four, ONE, two, ...", where the one is accented in the beginning of each 4/4 bar, can be seen as an implicit *topos* for quaternary classical music. A flutist reading her scores, knows she should, in general, accent the notes in the "ONE" position. This accenting creates a subjective rhyhtmic ostinato. As accenting is just a *choice* for her, finding the rhythmic ostinato "ONE, two, three, four" becomes a subjective task to the listener.

If we apply the concept of *topos* as "a periodic temporal reference" in this broader sense we obtain that both explicit rhythmic ostinatos and the traditional metric concept are particular cases of it. This generalization is not done by Kofi Agawu and will be adopted in the present work.

Another central issue found both in Agawu's and Fraisse's work or even in Lerdhal & Jackendoff's one is the importance of the body in the comprehension of rhythm. In this sense I shall comment the work [Iyer 02] where Vijay Iyer uses the Theory of Embodied Cognition to understand rhythmic phenomena. He argues that all musical rhythm experiences a human being can face in his or her life is related somehow to the rhythm of body functions such as breathing, heart beating, walking, speaking, etc.

His interest is to understand what he calls Micro-timing, small temporal deviations that emerges while a professional musician plays rhythmic cycles. His conclusion is that these deviations can only be conceived through Embodied Cognition and that they play a very important role on building a temporal reference between jazz musicians. This encourage me to add Micro-timing and in general, repeated asymmetric patterns as good building blocks of *topoi*.

Finally I cite a model that fits perfectly to the generalization of *topos* presented here. [Zbikowski 04] proposes a model of musical rhythm based on four propositions:

- P_1: Rhythm concerns regularly occurring events.
- P_2: There is differentiation between rhythmic events.
- P_3: Rhythmic events are cyclic.
- P_4: There is a strong sense of embodiment associated with musical rhythm.

He explicitly claims that he is not interested on creating an analytic model, instead, he sees it as "a guide for musical understanding" as follows: " Confronted with a sample of putatively musical sound we would look for manifestations of regularity (P_1) which were in some way differentiated (P_2). Were the sound only minimally differentiated a series of taps on a table top, for instance, or the steady drip of water from a tap we might suspect it was not music, and look for other things to clarify the situation. Among these would be some measure of cyclicity (P_3) that is, a higher- order pattern of differentiation which would group subsidiary patterns of more locally differentiated events. Finally, something that is really rhythmic, according to this model, is something we can, at the very least, tap our toes to (P_4)"

3.7 *Topos* Generalization

Based on all these references, I define a music piece contains (one or more) *Generalized Topos* (*Topoi*) if it contains (one ore more) *Recurring Pattern* (*Patterns*). The use of such non rigorous terminology is deliberate. I believe it is not necessary to define what are the possible *Recurring Patterns* that can be found in an audio excerpt, nor a precise test for doing so.

We can use [Zbikowski 04]'s work as an informal test for finding *Generalized Topoi* in audio information, but we are always subjected to wrong results such as [Jehan 05]'s experience since this test is culturally dependent.

It is important to notice that a recurrent pattern is detected only through a Gestaltian analysis. In fact, [Scheirer 00] points out taping cannot be done immediately, even by human subjects.

I include as instantiations of *Generalized Topoi*, musical phenomena such as:

1. The Cuban Rumba clave.
2. The surdo cycle in Brazilian Samba.
3. The meter counting of Concert Music.
4. The Macedonian odd meter's claves.
5. The Contemporary Jazz claves used by musicians to play odd meter songs.

I am not interested on developing a precise analysis of whether *Generalized Topoi* exist or not but the work [Foote & Cooper 01] seems to show a very effective way of computationally dealing with this subject.

In this work we shall notice an interesting phenomenon. The author depicts the beat spectrum of Paul Desmond's song "Take Five". In this analysis, the recurring elements create peaks on this spectrum's picture. The authors state that it is possible to see the recurrence of *triplets* which are a characteristic of the jazz *swing* accent.

On the other side the authors did not mention another important information that can be extracted from this picture: the use of a Contemporary Jazz clave.

Jazz musicians count "Take Five" using the following 5/4 clave: two dotted quarter notes followed by two quarter notes. That is the way the piano players accompany the piece and coincide with many accents of the melody. This fact is depicted in that beat spectrum. In fact, the 5-quarter notes recurrence is evident and coherent to the 5/4 clave period. The other two greater peaks are the one and the one-and-a-half quarter note recurrence which is coherent with the existence of this specific clave.

As we can see, these *Topoi* are not only an abstract and non rigorous concept, but can, sometimes, be subjected to clear computational analysis.

This discussion has been developed in an attempt to unify both the traditional and popular knowledge regarding Rhythm. No specific result is directly used in the rest of this text, but, for sure, this unified point of view was fundamental to the development of this research.

Chapter 4

Theoretical Framework

In the beginning of this thesis I listed and analyzed several examples of Interactive Systems. All of them have one or more specific algorithms to generate musical answers from their input. On describing them, their authors talk about different, "behaviours", "states" or "modes" created by these algorithms and that characterize the "types" of interaction the user face during the musical experience.

Each of these "modes of interaction" act as a "game" between the user and the computer. Although sometimes the relationship between the input and the output of these "games" is not clear, they are modeled by those algorithmic rules. Of course the game analogy cannot be taken too far because in the music interaction situation we do not necessarily have a winning goal that directs it.

If, on the other side, we focus our attention on real life situations like the Bat trio, the Gamelan ensemble (cited by Lewis) or the Steve Coleman's group, and if we link them to the improvisation analysis made by [Pelz-Sherman 98], and [Seddon 05], we arrive to the conclusion these "modes of interaction" are a common element to all those situations.

Based on this observation, I decided to guide my research through the practical path of building simple rhythmic-based command-driven "modes of interaction" that were joined together in Multi-Mode interactive systems.

These systems allow the performer to have the control of the Form of the piece. This means the narrative created during improvisation can be assisted by a variable-form strategy these Multi-Mode Systems provide, instead of forcing the performer to go through the traditional Sequential-Mode strategy.

To join these "modes" together, I used a concept coined by Francois Pachet: the "meta-commands", instructions given to the Interactive System that rule the management of the "modes of interaction".

In the sequence of this Chapter, I present a mathematical definition of "mode of interaction" using the Automata Theory. Then I develop the more complex concept of "meta-mode of interaction", a "mode" that controls "modes". This is done by improving the Synchronized Automata Theory with a Tree structure. Afterwards I focus on the formalization of rhythmic phrases that will be used to control these modes. Finally I give an overview of the problem

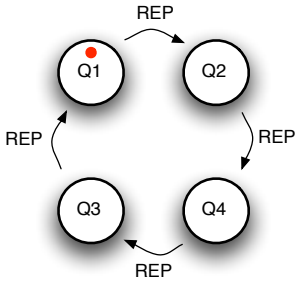

Figure 4.1: The vertices can be seen as the states of the automaton, and the letters in Q define the change of states. Here i_0 is signaled as a sign on the left side of the initial state.

above and my solution to it using this formalism. I fix the notation of the set $\{1, 2, ..., n\}$ as $[n]$ throughout this work.

4.1 Modes of interaction

The Automata Theory is shown to be a good foundation of the concept of modes of interaction both for its simplicity and rigour. As I am interested on studying interactive systems with many modes of interaction and as my main focus is on the interconnection between these modes, I need a model that deals with interconnected automata. [Beek 03] is my main reference to this subject and I use a slightly modified version of his approach.

Definition 1 *A deterministic finite automaton is a quadruple $A = (Q, \Sigma, \delta, i_0)$, where*

Q is the finite set of states of A
Σ is the finite set of actions of A such that $\Sigma \cap Q = \emptyset$
$\delta \subset Q \times \Sigma \times Q$ is such that if (p, a, q) and (p, a, q') are in δ, then $q = q'$; and
$i_0 \in Q$ is the initial state of A.

We can represent an automaton using an oriented graph such as in Figure 4.2.

We call $\{(p, q) \| (p, a, q) \in \delta\}$ the set of a-transitions of A denoted by δ_a.

I define a *mode of interaction* (also called *mode*) as a deterministic finite automaton. Each state Q represents a *musical context* in which the mode can be. The actions $a \in \Sigma$ define the possible transitions from the current state to other states. I define a *musical answer* of this mode to an action a as the generation or the modification of a certain sound the automaton produces. It worth noticing that to each musical answer corresponds a certain state transition (even if it is a reflexive transition from the state q into itself) but the inverse is not always true.

During the interactive experience, the musician can choose what musical answer he or she wants given the possibilities the music context offer and for that, apply the respective action that generates it. As this answer depends on the current context and on the action chosen by the musician, this means I use a *Mealy machine model*.

The the actions that cause these transitions are defined by the set δ. The initial state i_0 is the starting point of the musical interaction.

I am interested on studying interactive music systems with many modes of interaction so I need to deal with many automata at the same time. In the literature we find this concept of *synchronization* ([Beek 03, Ellis 97, Harel 87]) which is essentially the use of a certain type of Cartesian product of automata.

Consider S the finite collection $\{A_i | i \in I\}$ of automata, where $A_i = (Q_i, \Sigma_i, \delta_i, i_{i0})$, and $I = [n]$. To deal with the transition space of the automaton built over the collection S I follow the formalization given by [Beek 03]:

Definition 2 *Let $a \in \cup_{i \in I} \Sigma_i$. Then the* complete transition space *of a in S is denoted by $\Delta_a(S)$ and is defined as*

$$\Delta_a(S) = \{(p,q) \in \prod_{i \in I} Q_i \times \prod_{i \in I} Q_i \| \exists j \in I : \\ proj_j(p,q) \in \delta_{j,a} \wedge (\forall i \in I proj_i(p,q) \in \delta_{i,a} \vee proj_i(p) = proj_i(q))\}$$

We notice the abuse in the notation $proj_j(p,q) = proj_j(p) \times proj_j(q)$. Mainly, what is being said is that the transition space of the synchronized automaton must agree locally with the transition spaces of each automata in the collection S.

Now I can define the *Synchronized automaton.*

Definition 3 *A synchronized automaton over S is a quadruple $\Im = (Q, \Sigma, \delta, i_0)$ where*

$$Q = \prod_{i \in I} Q_i, \\ \Sigma = \cup_{i \ in I} \Sigma_i, \\ \delta \subset Q \times \Sigma \times Q \text{ is such that } \forall a \in \Sigma, \\ \delta_a \subset \triangle_a(S), \text{ and} \\ i_0 = (i_{01}, i_{02}, ..., i_{0n}).$$

This construction of Q as the Cartesian product of all the state spaces of each automata in the S collection gives freedom to create hierarchical structures of automata which is my interest here.

The general system I want to model has many predefined modes of interaction. I define them as the collection S above, each mode as an automaton A_i. Now I need a way to manage these modes.

I define the *meta-mode of interaction* (also referred to as *meta-mode*) as an automaton $M = (Q^M, \Sigma^M, \delta^M, i0^M)$. The idea behind it is that this automaton should control when each mode of interaction will be active or not. For this reason I need to adopt also a tree structure that relates these modes of interaction to the meta-mode.

We notice that M can be a synchronized automaton over the collection $\{M_j | j \in [m]\} = (Q_j^M, \Sigma_j^M, \delta_j^M, i0_j^M)$. This means that I can build hierarchical structure inside M before relating it to the modes of interaction. I opt to create just one tree structure that defines both the possible internal hierarchy of M and its relation to the modes of interaction as we will see further on.

Now I define my system $\Im$ as the synchronized automaton over $S' = \{A_1, ..., A_n, M\}$. As we noticed, M can be a synchronized automaton over $\{M_j\}$ and a typical state of $\Im$ will be of the form $(s_1, s_2, ..., s_n, (q_1, q_2, ...q_m))$ where $s_i \in A_i$ and $(q_1, q_2, ..., q_m) \in Q^M$. As the Cartesian product is an associative operation, we can see $\Im$ as the synchronized automaton directly over $\{A_1, ..., A_n, M_1, ..., M_m\}$. I am now in a position to define the tree structure of $\Im$. First I use a general definition of tree:

Definition 4 *Given the set T, the partial order relation $<$, and $t \in T$, we say $(T, <)$ is a tree if the set $\{s \in T | s < t\}$ is well-ordered by $<$. If T is finite, we define the height $h(t)$ as the number of elements of $\{s \in T | s < t\}$.*

Definition 5 *Let $(T, <)$ be a finite tree, the height $h(t)$ of $t \in T$, is the number of elements of the set $\{s \in T | s < t\}$.*

And I apply it to my case:

Definition 6 *Let $\Im$ be defined as above. I set $T = \{A_i\} \cup \{M_i\}$ and I define $(\Im, <)$ as the* Automata Tree*, a synchronized automaton together with a partial order relation on the set T that generates it. The elements of T are the* nods *of this tree.*

If $A < B$ for $A, B \in T$ and $\nexists C \in T$ such that $A < C < B$ we call A and B adjacent nods.

We define

- *i $\forall A_i$, $\nexists A \in T$ such that $A_i < A$, and*

- *ii if A and B are adjacent, we define there is one and only one state q of the automaton A which* represents *B.*

The function that defines this representation is

$$r : T \to \bigcup_i Q_i \cup \bigcup_j Q_j^M \cup \{\emptyset\}$$

where Q_i is the set of states of A_i and Q_j^M is the set of states of M_j.

We call r the representation function *of T and we say A is* represented *by q whenever $r(A) = q$.*

Finally we define the composition $r \circ r(A) = r(B)$, $A \in T$, where B is the automaton that has $r(A)$ as one of its states.

The hierarchy built in $(\Im, <)$ is used to define when a mode is active or not in an interactive situation. We should understand this tree in the following way: an automaton $A \in T$ is active if its parent $P < A$ is in the state $q = r(A)$. This means the actions of A will only be applied if the parent P is in the correct state q.

This can be stated as the *representation constraint* over $\Im$:

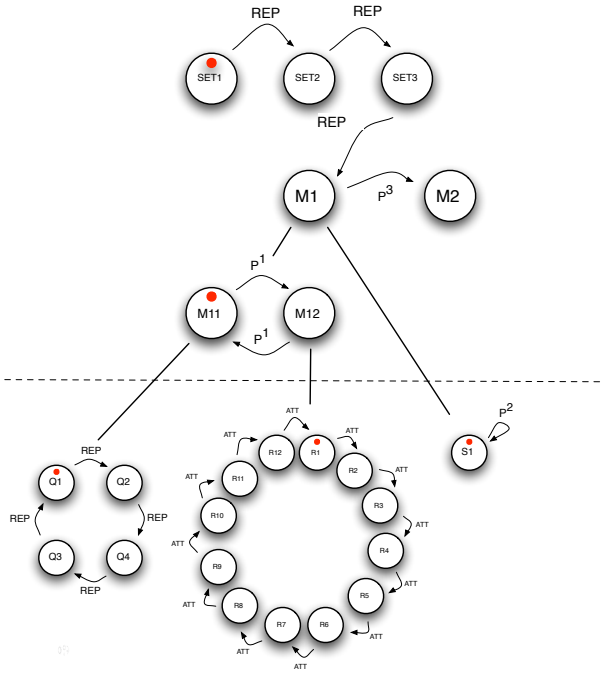

Figure 4.2: The vertices can be seen as the states of the automaton, and the letters in Q define the change of states. Here i_0 is signaled as a sign on the left side of the initial state.

$$\forall a \in \Sigma, ((s_1, ..., s_n), a, (s'_1, ..., s'_n)) \in \delta \Rightarrow \forall i \in I(s_i = s'_i \vee$$
$$\exists \text{ different } j_1, ..., j_k \neq i, k = h(A) : r^1(A) = s_{j_1}, ..., r^k(A) = s_{j_k}, s_i \text{ state of } A \in T).$$

As an example we see in 7.17 the automaton A built over $\{A_1, A_2, M\}$.

In item i I define the modes of interaction A_i must be the leaves of $(\Im, <)$. Each one is adjacent to a certain automaton in the collection $\{M_i\}$ and is represented by a state $r(A_i) = q_i^M \in Q_i^M$. This is because in the way I designed, two modes of interaction cannot be adjacent to each other and they must always be represented by a certain state of one of the M_j's. In this sense, a mode A_i will be on if it's adjacent automaton M_j is in the state $r(A_i)$ and off otherwise.

At this point, it's important to notice [Beek 03] does not use tree structures in his work. Instead, this approach is proposed by [Harel 87] although his scope is a visual representation of automata, what he named "statecharts", and he does not adopt this level of rigour.

Two other important concepts can now be defined.

Definition 7 *We say two modes of interaction A_i and A_j are excluding if exists $A \in T$ with state space Q and $k, l \in \mathbb{N}$ such that $r^k(A_i), r^l(A_j) \in Q$ and $r^k(A_i) \neq r^l(A_j)$.*

Definition 8 *We say two modes A_i and A_j are* transparent *if we can listen to the audio output generated by both of them together. We say they are* opaque *if the audio generated by A_i is stopped when A_j outputs its audio, in which case we say A_j is* over *A_i; and finally we say A_i*

filters A_j *if the audio from* A_i *is modified when* A_j *is turned on and comes back to normal when* A_j *is turned off.*

4.2 Rhythmic phrases

To control these modes of interaction, I propose the use of rhythmic information contained in the music signal generated by the user. My assumption is that the rhythmic information is easily and quickly analyzed from a monophonic audio source with small amount of error and is the first step in a possible melodic/harmonic approach to be developed in future works.

[Scheirer 98] argues that the rhythmic content of an audio signal can be extracted only from the amplitude envelopes of each frequency band of this signal. Here I will deal only with the peaks of these amplitude envelopes which I will call *attacks*.

I formalize the idea of extracting information from the audio signal in the following definition:

Definition 9 *The sign* r *is called* automatically extracted from the audio signal *if there is an algorithm that calculates a feature value* f_r *(typically an average) over the digitized audio signal (typically divided in sample buffers) and outputs this sign* r *whenever this feature is above a certain threshold* t_r*. We call* ϵ_r *the probability of this algorithm to correctly extract the sign* r *from the audio signal.*

An attack is an example of a sign.

Also, I need mathematical tools to deal with the sequence of attacks detected during the interactive experience. I shall notice this is the sequence of onsets and my analysis is based on the inter-onset intervals. As the length of this sequence cannot be defined in advance, I need a sequence of variable length.

Definition 10 *The* input sequence s *is defined as the function*

$$s : \mathbb{N} \times \mathbb{N} \to \mathbb{R}^+ \cup \{-1\}$$

such that given the current attack $c \in \mathbb{N}$*,*

1. $s(i,c) < s(j,c)$, $\forall i < j \leq c \in \mathbb{N}$
2. *if* $i \leq c \implies s(i,c) \in \mathbb{R}^+$ *and if* $i > c \implies s(i,c) = -1$
3. $\forall k \leq c$, $s(i,c) = s(i,k)$, $\forall i \leq k$

I denote $s_c(i) = s(i,c)$.

We say that the sequence s_c is the input sequence defined until the current attack c and the positions $s_c(i) = -1, i > c$ are said to be undefined. Typically, $s_c(i), i \leq c$ is given in milliseconds.

My interest is on the local information contained in part of the vector s_c.

Definition 11 *Given $k \in \mathbb{N}$, $k \geq 2$, we say $P_k \in \mathbb{R}^k$ is a* rhythmic phrase *(or just* phrase*) of length k if $\forall i, j \in \mathbb{N}$, $i < j \leq k$ we have*

$$P_k(i) < P_k(j).$$

This phrase can be collected from the input sequence *in which case we have*

$$P_k(l, s) = (s_c(l), s_c(l+1), ..., s_c(l+k-1)).$$

where s_c is an input sequence, $l \in \mathbb{N}$ and $l + k - 1 \leq c$.

We should notice that the input sequence by definition is also crescent. Depending on the context we can omit the index (l, s) and even k not to overload notation.

Finally I define a space to deal with these rhythmic phrases.

Definition 12 *We say ϕ_k is the k-*note rhythmic space *if it is the collection of all k-note rhythmic phrases $P_k \in \mathbb{R}^k$. We call $\phi = \bigcup_{k \in \mathbb{N}} \phi_k$ the* rhythmic space.

4.3 The solution

In this section I present a rigorous description of my approach using the tools developed above.

Let $(\Im, <)$ be an Automata Tree that represents an interactive system. Let $\Im$ satisfy the representation constraint 4.1. Our problem is to give the user access to the actions of this automata. The solution I propose is to associate to each action a_j, a sign r_j automatically extracted from the audio signal.

When this solution is implemented, the musician can use the sign r_j to switch the current mode of interaction. As r_j is automatically extracted from the audio signal, the user does not need to use any external meta-command to control the meta-automaton.

In the case studies, the actions and their associated signs will be called by the same name to simplify notation although I am aware they are two different things.

In practice, this sign can be the result of many real-time analysis over the input signal as we will see in the next chapter. As examples I can cite the attack sign, the detection of a certain pre-defined phrase or the detection of a repeated phrase.

Our hypothesis is that this method is effective to build a real-life man-machine interactive improvisation situation.

The comparison between this method and other possible solutions such as the use of pedals or other external meta-commands is not discussed here since the use of the audio as the only source of control is a premise of this work.

Chapter 5

Computational Issues

In this Chapter I present all the issues related to the implementation of the theoretical framework previously developed. A small discussion about the Sampling Theorem and Machine Listening issues is followed by practical considerations regarding the Pure Data, an implementation framework suited for real-time applications.

All the practical tools developed to deal with percussion-driven interaction are then elaborated.

5.1 Digital Sound

In the book [Bremaud 02], the author builds a mathematical framework that provides the principles of Digital Signal Processing. For sure the backbone of this Theory is the *Shannon-Nyquist Sampling Theorem*.

In that book, several versions of this theorem are presented. I state one of them (pp. 152):

Theorem 1 *Let $s(t)$ be a base-band (B) signal of finite energy. Then*

$$\lim_{N\to\infty} \int_{\mathbb{R}} \left| s(t) - \sum_{n=-N}^{+N} b_n sinc(2Bt - n) \right|^2 dt = 0,$$

where

$$b_n = \int_{-B}^{+B} \hat{s}(v) e^{2i\pi v \frac{n}{2B}}\, dv,$$

As in my case the musical signal can be considered continuous, limited, and of compact support, the theorem can be applied and the reconstruction formula takes the form

$$s(t) = \sum_{n\in\mathbb{Z}} s(\frac{n}{2B}) sinc(2Bt - n).$$

The hypothesis that a musical signal is base-band can also be considered true since, typically, the audition of an adult human being can capture sounds of at most 18KHz of frequency.

Other important results of this book are the formalization of Windowed Fourier Analysis, Digital Filtering, and the Fast-Fourier Transform (notated FFT throughout this text).

In fact, to deal with digital sound, these tools are not enough. The field of Psychoacoustics ([Zwicker 99]) models the several phenomena related specifically to human audition which must be taken into account.

An important work that relates Digital Signal Processing to Psychoacoustics and creates a Machine-Listening paradigm is [Scheirer 00]. My approach is indirectly based on this work since I use the Auditory Spectrograms of [Jehan 05] to build the low-level analysis of percussion signals.

The way I build my tools is now explained.

5.2 Pure Data Framework

The theoretical framework which is the core of this thesis, had to be implemented to produce real musical results. Several Digital Signal Processing computational frameworks directed to musical interaction in real-time can be found (eg. [Max], [Supercollider], [Chuck], [Puredata]). I chose the Pure Data framework and I use just the basic functionalities of this software such as audio acquisition and buffering, time management, data recording and output of midi information. All the core elements responsible for the interactive tasks developed in this chapter have been implemented in C language as Pure Data *Externals* (term given to the objects that are not part of the set of basic functions).

I use the brackets to define an object of this framework, eg. [timer], and the messages between these objects is notated with a [and a ¡ signs, e.g [bang¡.

One of the advantages of using this framework is that both audio and MIDI can be combined. This thesis is based on the use of audio signals to control the computer, and the MIDI information has not been used for this purpose. But, although sound synthesizing techniques were out of the scope of this research, they were, nevertheless, necessary for building the interactive experiences. In this situation I used the MIDI protocol as an output of my systems to send information to external synthesizers or other affine interaction systems.

Another tool this framework provide is the GEM, Graphical Environment for Multimedia, which is designed to the creation of audio-visual interactive experiences. I shall notice that one of the case studies here presented makes use of video output during the interactive situation using the GEM toolkit. The issues related to image-sound interaction will not be discussed in much detail because they are also out of the scope of my thesis, but will be pointed out in that chapter.

5.3 Percussion

As noticed in chapter three, the number of works on interactive music systems directed for percussion is not big. The main motivation for this thesis was to invest in this field for two reasons: the need of practical tools designed for percussion players to develop their interactive

experiences; and the comprehension that Rhythm has been underestimated or misunderstood by many people in the computer music community and the best way to shed some light on this subject would be through a percussion-driven research.

5.4 Instruments and Generalization

The research objects I show in this section were designed to deal with percussion instruments, more specifically the Pandeiro, which has been presented in the beginning of this thesis.

The generalization to other drum instruments is straightforward. Usually, the attacks of these instruments do not differ much from the ones I deal with, they are detected by a sudden increase of loudness in their audio.

Generalizing this low-level analysis to other instruments besides percussion is also feasible, the only modification to be done would be to add a real-time pitch tracker to the loudness analysis that makes my attack detection. This is enough to extract the onset of legato notes which can be treated as attack signs after being detected.

Besides this low-level stage, all the interactive tools here developed can be applied to other instruments since they are based on sequences of attack signs.

Informally, some musicians (a guitar player and a piano player) successfully tested the system. I omit this experience since is out of the scope of this thesis.

5.5 Low-level Analysis

The first analysis to be done in the audio signal is to detect the attack sign as mentioned in the last chapter. As I am dealing with real-time situations where time precision is needed, this analysis must be computationally light to avoid latency.

I decided to implement an attack detector (called [attack detector]) using a simplified version of Tristan Jehan's approach found in [Jehan 05]. In this work he first apply smoothing techniques to a windowed FFT analysis of the audio signal to model the temporal masking phenomenon of Psychoacoustics Theory obtaining what he calls the Auditory Spectrogram. From that, he extracts the derivative of the loudness function of each band and calculates peaks in these curves. An attack is extracted when a peak is stronger than a certain threshold.

Although my approach is based on these ideas, some modifications were made. The first difference is that I do not make the windowed FFT analysis on the input signal and consider the attack information I am searching for can be extracted only from the loudness feature of the whole signal. This can be done because I am not dealing with sustained sounds.

As I am dealing with real-time analysis, I obtain the digital signal in buffer frames of 64 samples (longer buffers of 128 or 256 were discarded to avoid latency). At each buffer, I calculate the average of the absolute value of the amplitude of these 64 samples. This gives a loudness value at each one millisecond approximately. I consider the set of these loudness values as the loudness curve of our signal.

I convolve this curve with a half-hanning window of 12.5 milliseconds to obtain a smoother

curve still sensitive to abrupt changes of loudness. [Jehan 05] signals the fact that the convolution with a half-hanning window of 50-ms would model the temporal masking phenomenon and, by doing so, we would arrive to a result closer to what human beings can hear. In my situation many values were tried and 12.5-ms seemed to be the best compromise between latency and robustness, which is my focus.

Continuing my process, I differentiate this convolved loudness and obtain the derivative curve. Finally I convolve this curve with a Hanning window of 12.5 (as does Tristan except for the different window length).

Our attack detection follows two steps. First the algorithm checks, each time a buffer is delivered, if the convolved loudness curve is greater than the *loudness threshold*. If it is, the algorithm checks if the convolved derivative is greater than the *derivative threshold*. If both conditions are satisfied, the algorithm reports the detection of an attack. During a small amount of time (50-ms), the attack detector, as is called the algorithm, stays disabled because the convolved derivative can still be greater than its threshold after the detection. In fact, the convolved loudness curve can grow much more than as it does in this initial instant depending on the strength of the attack, but my interest is to detect it the fastest I can.

[Jehan 05] is concerned with offline analysis and instead of comparing the convolved derivative with a threshold he searches for peaks on this curve. This would not be possible to us because we would need to wait the whole increase and decrease of the convolved loudness curve to report an attack and this would mean a bigger latency. In my case, the only intrinsic latency of this process comes from the convolution of the derivative curve with the Hanning window. Because of this, the information I have in the convolved derivative curve at time t is related to the information of the derivative curve at time $t - 6.25$-ms were the Hanning window has it's peak. As the derivative curve is too noisy and as 6.25-ms is imperceptible, I chose this method.

As I am designing interactive systems to work on stage, the background noise (which is sometimes much stronger than just a background) must be taken into account. That's why I decided to design a calibration phase previous to the use of the attack detector.

The user is asked to let his or her microphone "opened" to capture the loudest background noise possible for a period of time (typically 10s). During that, the system calculates the maximum value of the convolved loudness curve and stores it in the variable 'MaxLoud'. It does the same to the convolved derivative curve storing the result in 'MaxDeriv'. The loudness threshold is defined as $1.05 * MaxLoud$ and the derivative threshold as $0.8 * MaxDeriv$.

Firstly I shall explain why I use two thresholds and secondly why they are defined like that. The reason is I am interested on being able to detect very fast sequences of attacks, and to distinguish each one of them. When two attacks are very close in time (eg. 100ms) the attack of the second sound is summed up with the reverberance of the previous one. If I used just the loudness threshold, the second peak would not be detected unless the loudness curve would have had the time to decrease and cross this threshold again.

On the other side, due to the reverberation of the first attack, the growth of the loudness curve caused by the second attack could have not been enough to cross the derivative threshold a second time. So, it is necessary to use a smaller value than 'MaxDeriv' but using just that would cause many mistakes, by definition.

Empirically I arrived at the value $1.05 * MaxLoud$ to give a small margin to the calibration value MaxLoud and at $0.8 * MaxDeriv$ because of the reverberation phenomenon described above.

As said before, the Pandeiro events I am interested in are the 'pa' and 'tung' attacks and the 'tchi' sounds are heard as background noise. To ensure the attack detector picks up only the right sounds, the user can play a series of 'tchi' sounds during the calibration phase. Doing so, these sounds will not make the loudness curve cross its threshold and will not be recognized as attacks.

A few rhythmic tools I developed use the 'tchi' sound as input information in which case the calibration shall be done only with real background noise.

These practical issues were also a reason to design the calibration phase.

Although a comparison is out of the scope of this thesis, I informally tested the *bonk~* object designed by Miller Puckette ([Puckette 98]). The latency on detecting attacks was uncomfortable for dealing with the precise rhythmic tools described further on. Also, the calibration phase is not clear in the documentation or does not exist. For these reasons I decided to develop and implement my own attack detector.

5.6 Rhythmic Phrases

As we know the rhythmic phrases assume a fundamental role in this work. I developed in the last chapter the formal concept of variable length sequence to deal with these attacks.

In the computational level, these sequences will be an enoughly long vector called a. Dynamic memory allocation could have been used but in a typical interactive situation the number of attacks usually do not exceed one hundred thousand and as nowadays computers allow, I always allocate one million floating points to a.

In figure 5.1 we see the representation of the attack detection and storage. $a[i]$ stores the instant this attack happened in milliseconds. This is done in Pure Data using a [timer] object. The [attack detector~] receives the sound from the audio input [adc~] and sends a [bang¡ message when an attack is detected. The [timer] calculates the time spent from its initialization (which happens when loading the patch) to this last [bang¡ and send this information to whatever interactive object I developed. This object always allocates the a vector as I said before and records at position $a[i]$ the instant in milliseconds this attack happened.

We noticed in the last chapter that the rhythmic phrases I deal with are a sequence of onset intervals whose boundaries are the attack signs. This means that in my case the final boundary of a rhythmic phrase is the first attack of the next phrase. By implementing other kinds of low-level analysis I could define also the *duration* of each note independently of the placement of the next attack, but this is not necessary for the tools I developed.

When needed, I will point out these boundary issues in what follows.

At first glance, the choice of representing the input musical signal using just the vector a can seem too simplistic. Working on experiments, it became clear that this representation carries the essential information to deal with complex interactive situations. The results I present in this thesis prove that assertion.

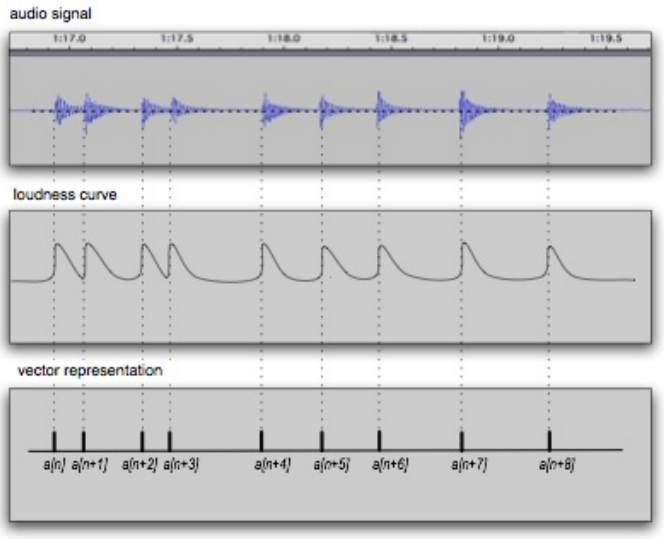

Figure 5.1: The audio signal has its loudness curve analyzed. The attack signs are detected and stored.

5.7 Distances

My first task on working with the input sequence is to define a way to compare rhythmic phrases. There are many possible ways of doing it. During my research I developed some methods based on [Foote & Cooper 01]. Using these measures it is possible to develop some signs automatically extracted from the audio signal. I now summarize the most important signs developed in this research.

5.8 Signs

Now I can define the signs which were implemented and used in the design of my systems. Some of them can carry information to be used as a parameter in the interaction situation.

5.8.1 Attack

As we saw in the low-level analysis section, the first and more basic sign I deal with is the attack. The explanation on how it is extracted has already been given.

An important information it can carry is the intensity of attack. In fact, the [attack detector] object I developed has also an intensity calibration phase that was omitted in the last section. During this phase, the user is asked to play his or her instrument the loudest possible and the maximum value of the convolved loudness curve is stored. When an attack is extracted, the system outputs the instant value of the loudness curve divided by the maximum value captured in the calibration phase.

Of course this value can exceed 1 but generally it stays in the [0 1] range.

This value carried by the attack sign can be used in a series of situations, for example to define the volume intensity of a sampled sound played by the machine.

The action to which the attack sign is associated is called ATT.

5.8.2 Phrase Detection

On several of my experiences there is an important sign that has been used: the detection of a certain phrase in the input sequence. This detection is done using the above mentioned distance measures.

5.8.3 Repetition Detection

The importance of repetition in Music is obvious. The presence of repeated forms in classical and popular music is ubiquitous and usually the audience and the musicians can distinguish between periodic and non-periodic parts of a piece.

I chose the detection of repeated phrases as a mechanism to control a series of functionalities of my interactive experiences.

5.8.4 Silence

A sign worth noticing is the Silence Sign. It can seem strange that silence could be used as a control, but in fact is one of the most important and natural ones. Defining silence can be a hard philosophical task but the common sense says silence is "the absence of sound". In my case I interpret it as the absence of attacks. But the attacks are dimensionless so there is always an absence between two of them. I define a Silence sign when there is an absence of more than n milliseconds after an attack.

This can be implemented using a Pure Data object called [delay n]. This object sends a message after n milliseconds it received the last message. If we link the [attack detector] with it, each time an attack is detected, the [delay n] is reset and, if no attack message is received after n milliseconds, it sends a Silence sign:

[attack detector]
|
[attack<
|
[delay n]
|
[silence<

An important use for this sign is when defining multiple parts of a piece, as we will see in later on.

5.8.5 Attack Classification

As mentioned in the background chapter, the field of Music Information Retrieval provides many techniques to classify, in real-time, the distinct sounds a percussion instrument can produce. In the case of the Pandeiro this has been solved by [Roy, Pachet & Krakowski 07/2] using the EDS system at Sony/CSL-Paris in a research I collaborated. As said before, although this result has been implemented, I decided to focus on questions raised by the use of attack detection letting the sound classification as a step for future work. On the other side, I had already developed a very simple classifier that has been used in some of my experiences and which I now describe.

By observing the very first 64-sample buffer of the Pandeiro's 'pa' and 'tung' sound attacks, I realized the 'tung' attacks generally presented a very high frequency (around 20KHz) not found on the 'pa' attacks. Although this is an ad-hoc method, it was useful to develop a 'pa-tung' classifier used in my experiences. This classification has the advantage of being based on just the very first 64-sample buffer of the sound attack, which is the lowest latency I could have.

Chapter 6

Experiences

To deal with Music means to deal with aesthetic choices. Whatever research in the Computer Music field that has any compromise with real Music also is attached, at least, to aesthetic motivations. Some of these researches validate their result objectively, without depending on any aesthetic judgement, which is the case of many MIR works.

On designing Interactive Music Systems, this neutral validation can be more difficult. What makes a system valid or better than the other? In my case, where interaction is rhythmic-based, the situation is even more difficult because of the lack of related works.

The validation I chose was the most objective possible: to be able to play improvised pieces (about 3 to 5-minute long) designed to percussion and computer in an on-stage real-life popular music situation.

But in my point of view, the most valuable goal of this thesis are the questions raised during the process of research. For this reason I give now a chronological overview of the interactive experiences I developed and the aesthetically based choices that led them. I believe this narrative approach can be useful to give a better view of the whole. Also, some "narrative" works like [Jorda 02/2] and [Cook 01] were useful to me and I hope this work can be useful to other people.

6.1 Early Experiences

The very first experiences were done with a set of modes of interaction implemented as disjoint Pure Data objects. The idea of building a piece with these mode was still far from my reality and I could interact with one mode at a time.

The very first mode I built was called [Copier] and worked in a very simple way: I played two notes setting up the length of a beat and then, a beat later, I played a 2-beat long rhythmic phrase that was recorded and looped by the object using a synthesized sound.

Another object I developed at that time and that still interest me is the [PlayQuant]. It worked in the following way: I could play freely and when I repeated a phrase, the object recorded that phrase and quantized it. The quantization was done by searching which grid best matched the tatum of the phrase. As the grids ranged from 3-note to 17-note length, sometimes a 4/4 phrase was quantized in a strange grid of 15 notes, for example. After the object recorded

and quantized the phrase, it looped this quantized version and I could play over it. When I decided to record another phrase, I just had to repeat it twice and the object substituted the looped phrase by the new one.

The other object I created at that time was the [Layers], which has been used in the first case study I present here. There we find a clear explanation on how it works.

Although these objects present very simple interaction modes, I think they open a path to automata-driven rhythmic interaction which I hope I can develop in future works.

Some of the sonic results of these experiments were interesting, as we can listen in [Early Experiences], but, at that point, a question anguished me: How could I make a musical piece using these interactive tools?

6.2 Aguas de Maro

I turned to this other challenge of trying to play a musical piece using computer and pandeiro.

I chose the tune "Aguas de Marco", by Tom Jobim. The arrangement I have done featured a singer accompanied by pandeiro and computer, which was responsible for the whole harmonic background.

To accomplish that task, I had to develop several tools including a beat-tracker that followed me in certain portions of the piece where the computer played the synthesized harmonic accompaniment.

On a central part of the arrangement, I substituted the classic flute bridge by a pandeiro solo where I used the *Layers* object to create a percussive atmosphere.

Each part of the piece was pre-defined, some were more interactive such as the *Layers* part, others less, such as the "score-following" part.

The sequence of the parts was also pre-defined (Sequential-Mode paradigm), I did not have the option to switch from on mode to the other. The choice I had was on the length of the *Layers* part, which finished only by a Silence sign.

I performed this piece twice, for video recording ([Aguas de Marco]) and without audience. It was impracticable to play it 'on stage' due to the way it was implemented. The challenge of building a whole piece with pandeiro and computer was accomplished, and it became clear the approach had to be improved.

6.3 Drum-Machine

In march 2007, I had an invitation from the Oldham Percussion Festival to do a pandeiro-computer performance. It was clear that the strategy used in the Agua the Marco piece could not be repeated there, so I decided to use an object I had recently developed: the [phrase detector].

The whole description of this system and the two musical examples are given in the Drum-Machine case study.

At this point it became clear that, using a Multi-Mode paradigm, it was possible to go 'on stage' with a flexible system that allowed me to create improvised pieces on-the-fly.

6.4 Rhythm and Poetry

An artistic need I had was to deal with the words a as sonic material. That is why I developed the two pieces [RAP] and [Cabecas].

There I came back to the Sequential-Mode paradigm but the implementation allowed me to perform the piece in a real-life situation. I did not use the beat-tracking technique. I had the choice on when to change from one part to the other and this was done using some of the signs mentioned in the last Chapter

6.5 Pandeiro Funk

In July 2008 I decided to create the ChoroFunk concert series and one of the important parts of it was the Pandeiro Funk solo developed in the middle of each concert.

The system and its results are discussed in the second case study of this thesis.

There, it became clear the importance of the Multi-Mode architecture to live performances. This conclusion did not happen by hazard, I tried several paradigms on real-life situations as we see by the description of these experiences.

This long path is what made the final results of this research robust to 'on stage' situations.

6.6 Other Results

Many other partial or complete experiences have been developed. Some of them include other musicians. I informally tried the Multi-Mode systems to be controlled by a guitar and a piano and it did work. I also developed some systems able to record and use the audio of other instruments live.

Another important result I obtained was the final work of the course on Statistical Learning taught by professor Paulo Cezar Carvalho at IMPA. I developed a group of seven simple feature functions calculated over rhythmic phrases that were analyzed by several statistical classifiers. SVM turn out to be the most efficient one.

This opens the possibility of using other kinds of sign, more complex, that can make Rhythmically-Controlled Interactive Systems more flexible.

Chapter 7

Case Studies

Many case studies have been developed during this PhD research. As I did not find any other work in the literature dealing with multi-modal systems and rhythmic interaction, it was not possible to develop a comparative study between our results and the others'. Because of that I chose to build an experimental research where many new rhythmic interaction modes were developed and implemented and where the *switching mode problem* arose as the main issue to be solved. I present now two of the seven case studies developed during this research.

7.1 Drum-Machine Case Study

I discuss here a case study of a system designed to interact with the Pandeiro (Brazilian tambourine) that has three modes of interaction and uses the tools defined above. Two of these modes use the Continuator system ([Pachet 02]) to produce harmonically interesting piano chords. All three modes and the meta-mode that controls them is now described in details.

As I exposed before, my problem was to be able to switch from one mode of interaction to the other in a natural way during the interactive experience. The solution I proposed was to use signs automatically extracted from the audio signal. I present here a system that is a practical example of this solution, the signs I choose are the detection of certain rhythmic phrases.

Musically, the main goal of this case study was to allow the musician to build drum-machine loops and to improvise over this rhythmic base whether by using the natural acoustic Pandeiro sounds or Continuator chords released at each Pandeiro attack.

First I describe the architecture of the system using the formalism already defined. Then I show a graphical representation of two musical pieces developed using these systems and comment on the resemblances and differences between them.

7.1.1 System Architecture

The three modes of interaction are described here as well as the meta-mode that controls them.

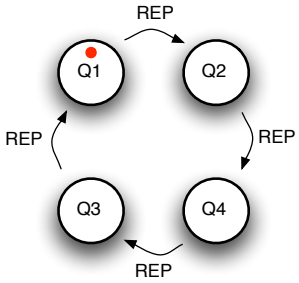

Figure 7.1: Each state represents a voice to be recorded. Every time a phrase is repeated, the automaton starts to loop this phrase and changes its state to be prepared to record the next voice.

Drum Layers Mode

The first mode of interaction is a user-programmable drum machine. The concept of drum-machine dates back to the 30's and became popular in the 80's with the Roland TR-808 machine. The purpose of these machines was to create drum kit grooves using synthesized or sampled sounds. The basic idea is to divide the drum loop in voices. Each voice can be seen as a separate one-instrument loop that is summed up to form the whole groove.

The Drum Layers Mode allows the player to build a whole drum-machine loop by programming each voice without external meta-commands. The musician uses the phrase repetition sign to do that. In this case study, the instruments chosen to be the loop's voices were: the bass drum, the snare drum, the hi-hat and the iron triangle.

The automaton that represents this mode is depicted in figure 7.1. I call this the automaton $A_1 = (Q^1, \Sigma^1, \delta^1, i_0^1)$. As we see in that figure, this automaton has four states $\{Q1, Q2, Q3, Q4\} = Q^1$ and just one action $\{REP\} = \Sigma^1$. This action is associated to the repetition signal previously defined. Each time this sign is received, the automaton leaves a certain state and builds one of the voices of the drum machine loop.

I now explain how it works. In the initial state $Q1$, the automaton does not produce any audio output. If the user repeats whatever phrase P twice, the algorithm that detects repeated phrases in the input sequence will generate a repetition sign which makes the automaton switch to state $Q2$. When it leaves state $Q1$ it produces the bass drum voice. This voice is an audio signal that is looped many times (further on I explain when this loop stops).

The automaton is then in state $Q2$ and is prepared to record the snare drum voice. When the user repeats another phrase P' the automaton changes to state $Q3$ and generates the snare drum voice exactly as happened in the bass drum case. The bass drum and the snare drum voices are summed up and the user can listen to both of them. It is important to notice that the voices need not to be of the same length. This allows the user to create polyphonic effects. The algorithm that manages the building of the voices adjusts their length not to occur a desynchronization phenomenon.

Again, when the user repeats another phrase P'', the automaton leaves state $Q3$ to state $Q4$ and produces the hi-hat voice. If the user repeats the fourth phrase P''', the iron triangle

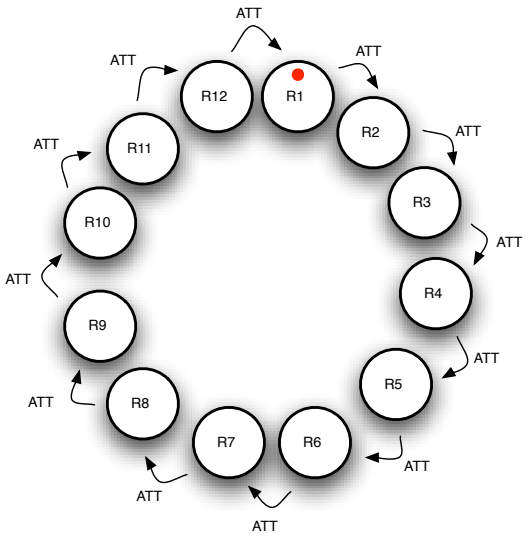

Figure 7.2: Each state represents a note to be played. At each attack sign, the automaton plays this note and goes to the next state.

loop will be recorded and the automaton will go back to state $Q1$. At this moment, the bass drum previously recorded is deleted and the user can record it again. When the user passes to state $Q2$ the new bass drum is recorded, the snare drum voice is erased and so on. This way the user can progressively change the content of each voice which creates a dynamism of the drum machine groove.

Short Chords Mode

This mode of interaction permits the user to release short duration (0.2s long) piano chords simultaneously to each attack produced by the pandeiro.

The chords are not produced by my system directly, instead, they are produced using the*Continuator system* in the harmonizer mode. In this mode, the Continuator receives a note (using the MIDI protocol) and chooses the best chord (also represented using MIDI) that harmonizes it. This judgement is done with respect to the musical style the system learnt from the player in a previous stage. As the Continuator uses this style dependent criteria to harmonize each note, it's possible to listen to a harmonic coherence in the sequence of chords released by the pandeiro attacks in the musical examples of this case study. That was the motivation to use the Continuator instead of, e.g., a random choice of chords.

Our system have to decide what note to send to the Continuator at each pandeiro attack. In this case study I used two strategies of choice (one to each piece here presented): a pre-defined sequence of twelve notes and a pseudo-random choice in a chromatic scale.

First I present the strategy of the pre-defined sequence. The automaton A_2 used to model this choice is depicted in figure 7.2. Each state $R1$, $R2$, ..., $R12$ represents a note of this pre-defined melody. The action ATT is associated to the attack sign. This means if the automaton

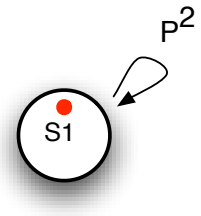

Figure 7.3: Each time a P^2 action is applied, the automaton releases a long chord and comes back to its unique state.

is in state $R1$ and the player produces an attack sign, the automaton sends note 1 to the Continuator to be harmonized and goes to state $R2$. The chord generated by the Continuator is sent back using the MIDI protocol and is synthesized using Hammond piano sounds. Ideally, this synthesized chord should be heard at the same instant we can hear the pandeiro attack, as if they were "attached" to each other. In my case, there is a small delay (around 30 ms) between the attack and the chord release that is barely perceptible and, thus, can be ignored.

Then, when the system is in state $R2$ and the user produces another attack, the second note is sent to the Continuator, harmonized and a new chord is synthesized again. The automaton goes to state $R3$ and so on. When it arrives at state $R12$, it goes back to state one and restarts this twelve notes cycle. Although the melody sent to the Continuator is cyclic, the harmonization changes each time this cycle is repeated, which gives a richness in terms of harmonic paths, even though the sense of cyclic melody is still preserved. This is another motivation to use the Continuator as a harmonizer. All that will be observed in the musical examples presented further on.

The second strategy is to choose randomly whether to go up or down in a chromatic scale. We omit the figure that represents this automaton, but it suffices to say it has only one state and one action ATT that links this state with itself. Every time the user produces an attack sign, the automaton sends a note to the Continuator and comes back to this state. This note can be either one semitone higher (.5 of probability) or lower (.5 of probability) than the note sent in the last attack sign. As will be observed, in this case, the melodic sense is absent, but the harmonic coherence is still preserved.

Long Chords Mode

This mode is similar to the Short Chords Mode. It allows the user to release a long duration (2s) piano chord.

Figure 7.3 shows the automaton A_3 that models this mode. As can be seen, there is only one state $S1$ and the action P^2. This action is associated to the detection of phrase P^2 from the input sequence. This is a case where the action receives the same name of the sign it is associated to. Further on I will explain how does this phrase is defined. Every time this phrase is detected in the input sequence, my system sends a C note (MIDI note number 72) to the Continuator which harmonizes it. This chord is synthesized as before, the only difference is that

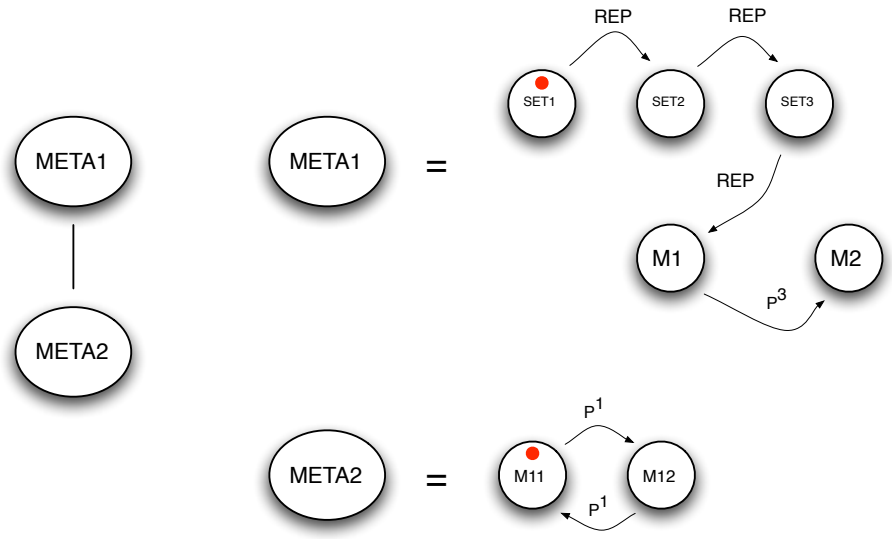

Figure 7.4: Tree structure of the meta-mode automaton that controls the system. The two meta-automata that form the meta-mode are graphically defined on the right of this picture.

the duration of its sound is two seconds long instead of the two hundred milliseconds of the Short Chords Mode. The other difference from this mode to the Short Chords is that here the chord is played "immediately" (the latency can again be ignored) after the last note of phrase P^2 is detected, in the previous case, the chord was released after each attack was detected.

Meta-mode

The role of the meta-mode is to model the meta structure that allows me to work with all these modes of interaction in the same musical system.

As said before, the meta-mode can have a tree structure itself. In this case the tree has just two nods, $META1$ and $META2$, depicted in 7.4. A rigorous description of the Automata Tree $(\Im, <)$ that models the entire system will be given in the next section.

In figure 7.5 I depict the automaton that represents the meta-mode. We should notice the tree structure presented above also appears here connecting the automaton $META2$ to the state $M1$. This happens because, as we will see further on, $r(META2) = M1$.

As I previously said, the signs chosen to control this system are the detection of rhythmic phrases. These phrases are not predefined, this means the user is able to choose them in a setup phase previous to the musical interaction itself.

This phase is represented here by the states $SET1, SET2, SET3$, and they will be responsible for the recording of the phrases P^1, P^2 and P^3 respectively.

We notice the action REP is found again in this automaton. This action is also associated to the repetition sign as before, but here its utility will be different from the drum machine case. When the meta-automaton is in the initial state $SET1$, if the user repeats a certain phrase, this phrase is recorded as P^1 and the meta-automaton changes to state $SET2$. No musical answer

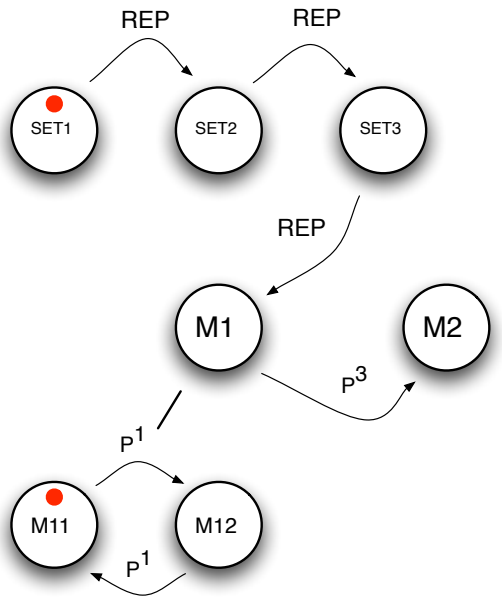

Figure 7.5: Complete meta-mode automaton including tree structure and explicit description of automata $META1$ and $META2$. Notice $META2$ is represented by $M1 = r(META2)$, a $META1$ state

is generated at this transition, that's why I call it setup phase. Again the user repeats another phrase which is recorded as P^2 and the meta-automaton passes to state $SET3$. Finally the last phrase P^3 is recorded in the same way, the meta-automaton leaves the setup phase and passes to the meta-mode $M1$.

At this point the meta-automaton enters the interactive phase. As we will see later on, the state $M11$ represents the drum layers mode, $M12$ represents the short chords mode, $M1$ represents the long chords mode and $M2$ is the end of the interaction. This means that, if the user is in state $M1$ he or she can choose to go from the drum layers mode to the short chords mode or backwards using phrase P^1, to release a long chord using phrase P^2 or to finish the piece using P^3. A detailed description of an interaction situation will be given further on.

It's important to notice the transitions of this meta-mode do not generate any sonic result except for the transition $(M1, P^3, M2)$ that causes the system to stop all the sounds that are being generated and play a sequence of chords followed by a cymbal sound. This is the ending of the piece and there is no way the user can leave the state $M2$ as we can see in the picture 7.5. That's why I defined it as the end of interaction.

Now I rigorously define the meta-mode as the synchronized automaton $M = (Q^M, \Sigma^M, \delta^M, i0^M)$ over the collection $\{META1, META2\}$ where

$$META1 = \{SET1, SET2, SET3, M1, M2\} \times \{REP, P^3\} \times \delta^{META1} \times \{SET1\}$$
$$META2 = \{M11, M12\} \times \{P^1\} \times \delta^{META2} \times \{M11\}.$$

The sets δ^{META1} and δ^{META2} that define the possible transitions of these automata can be

inferred from picture 7.5 and are omitted here.

Explicitly,

$$M = (\{SET1, SET2, SET3, M1, M2\} \times \{M11, M12\},$$
$$\{REP, P^1, P^3\},$$
$$\delta^M, (SET1, M11)),$$

where again δ^M is inferred from picture 7.5.

Before building the Automata Tree that models the whole system, it's important to notice the role of the states $M11$, $M12$ and $M1$ is to represent respectively the drum layers mode, the short chords mode and the long chords mode. This means each one of these modes will be active only if M is in its respective state.

The Automata Tree

Theorem 2 *Let $T = \{A_1, A_2, A_3, META1, META2\}$ be the collection of automata defined in this case study.*

Let $<$ be a partial order in this set defined by

$$META1 < A_3, META1 < META2, META2 < A_1 \text{ and } META2 < A_2.$$

Let $\Im = (Q, \Sigma, \delta, i0)$ be the synchronized automaton over the collection T.
Let $r : T \to \cup\{Q_1, Q_2, Q_3, Q^{META1}, Q^{META2}, \{\emptyset\}\}$ be defined by

$$r(A_1) = M11, r(A_2) = M12, r(A_3) = M1, r(META2) = M1 \text{ and } r(META1) = \{\emptyset\}.$$

Then, $(\Im, <)$ is an Automata Tree.

Proof 1 *It suffices to show that items i and ii of 6 are satisfied.*
Item i: in fact, $\forall i \in [3]$, $\forall A \in T$, if A is comparable to $A_i \Rightarrow A < A_i$.
Item ii: given $A, B \in T$, if they are adjacent, $r(B) = q$ where q is a state of A.

We can see the tree $(T, <)$ depicted in 7.17.

If none is said, as $\Im$ is a synchronized automaton, δ has only to satisfy the condition $\delta_a \subset \triangle_a$ which means the local transitions of the automata in T are preserved. But this does not ensure that the representation function is doing its role of creating a dependence between each automaton in T and the state that handles it active. To this purpose I force $\Im$ to satisfy the representation constraint 4.1.

In figure 7.7 we see the Automata Tree $(\Im, <)$.

This picture should be read in the following way:

The red dots define the current state of $\Im$. The picture shows the initial state $i0$. When the user generates a sign, the system checks if the root automaton accepts the action associated to it. If it accepts, the red dot of the root automaton leaves its state and follows the arrow of this action to enter its next state. If this automaton does not accept this action, the system checks if the red dot of the root automaton is in a state that represents another automaton (in

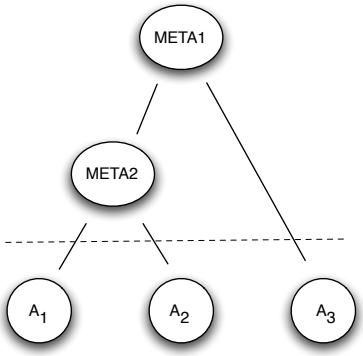

Figure 7.6: Tree structure T of the whole system. Above the doted line we see the meta-mode automaton M, below it we see the modes of interaction

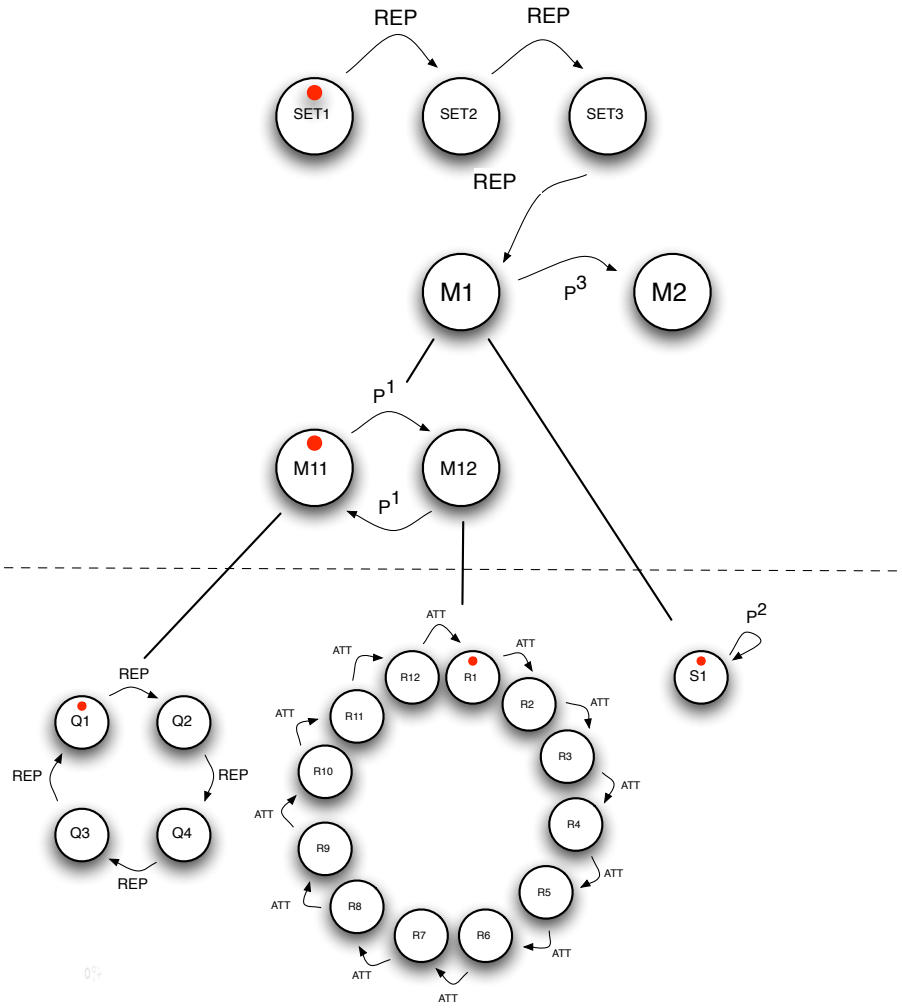

Figure 7.7: Automata Tree $(\Im, <)$ that models the system of this case study.

which case there will be an edge of the tree attached to it). If it does not represent any other automaton, the action is ignored, if it does represent an automaton, the system checks if this automaton accepts the action. This is done recursively until arriving to the leaves of the tree, the modes of interaction.

Observing figure 7.7 there are some important structural facts that can be extracted.

First, all the SET states that correspond to the setup phase above mentioned do not represent any automata, so the only action accepted by $\Im$ during these states is REP until $META1$ arrives at state $M1$. If, at this stage, action P^3 is taken, the interaction ends because $META1$ goes to state $M2$ from which it cannot leave (because there are no actions from it to another state). This means the whole interactive phase happens when $META1$ is in $M1$.

$M1$ represents $META2$ (left edge) and A_3 (right edge). By its turn, $META2$ represents A_1 (left edge) and A_2 (right edge). But the state that represents A_1 is $M11$ and the one that represents A_2 is $M12$. This means they are excluding or non-orthogonal, the user cannot play with the drum layers and the short chords at the same time. On the other side, A_3 is orthogonal to both states, in other words, it is active in spite of the fact that A_1 or A_2 are active.

The graphical representation 7.7 informs even more. As said before, the initial state $i0$ is represented by the red dots; the tree structure $(T, <)$ is explicit, all the actions Σ and the transitions δ are evident; finally, the explanation on how the picture should be read forces the system to satisfy the representation constraint. Because of that, we can say this graphical representation *completely defines* the structural aspects of the interactive system (the Automata Tree $(\Im, <)$ and the representation constraint).

On the other side, this picture does not inform about how the sonic results of one mode affect the others'. In this specific case we shall notice the drum layers mode and the short chords mode are transparent, but the long chords mode is over them.

To make these issues clearer let's build an example.

Example The picture 7.7 shows that the initial state of $\Im$ is $(Q1, R1, S1, SET1, M11)$. The current state of the root automaton $META1$ is $SET1$ and this state does not represent any other automaton because there are no edges of the tree that are connected to it. So, this means the only possible action $\Im$ can accept in this situation is REP. Let's say the user repeated a certain phrase P^1. As explained before, this phrase is recorded as a sign and $META1$ changes to state $SET2$. This will be represented by changing the red dot from $SET1$ to $SET2$ and the current state of $\Im$ will be $(Q1, R1, S1, SET2, M11)$. As $SET2$ does not represent any automaton neither (there are no edges of the tree connected to it), again the only action accepted is REP. Let's say the user repeated the phrases P^2 and P^3 leaving the setup phase and arriving to the state $(Q1, R1, S1, M1, M11)$. From now on the user will be in the interactive phase.

In figure 7.8 I depict the current situation.

As $META1$ is in state $M1$, the system accepts the action P^3. But as $M1$ represents both $META2$ and A_3, the system accepts respectively P^1 and P^2. Finally, as $M1$ represents $META2$ and as $M11$ represents A_1, the system also accepts REP.

Let's consider what does it mean each one of these possibilities:

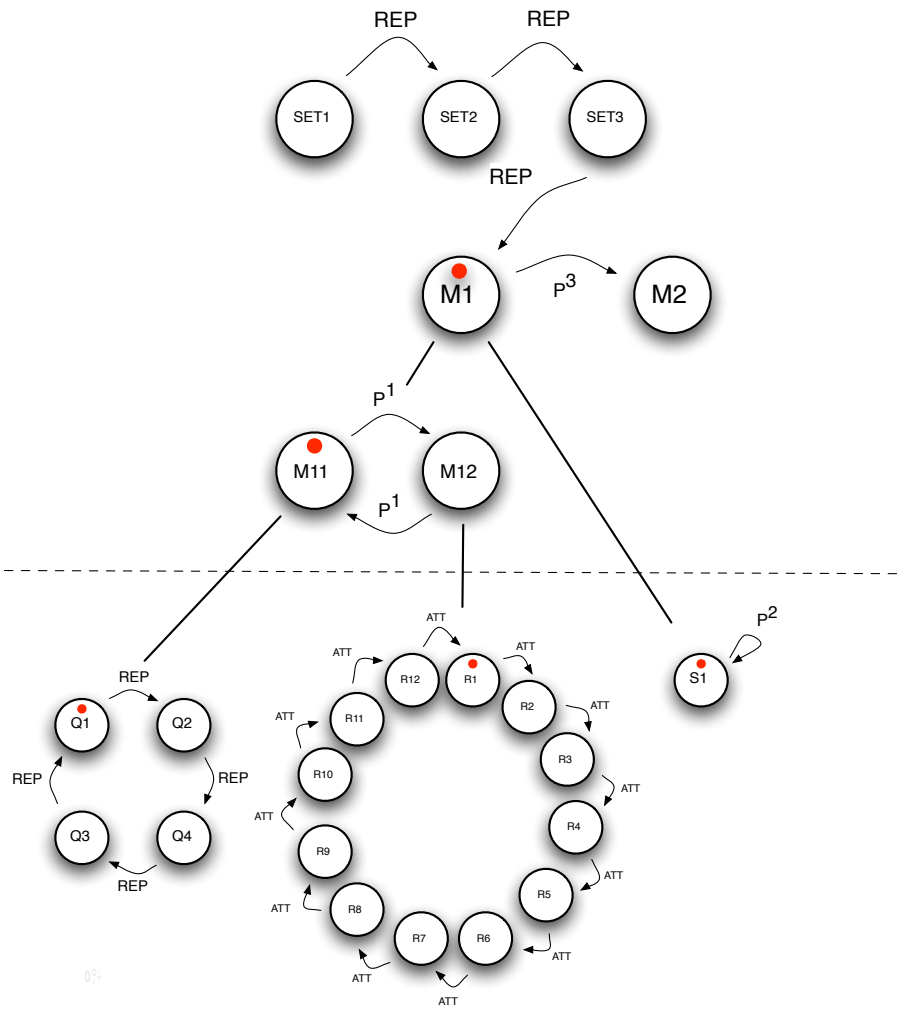

Figure 7.8: $\Im$ in the state $(Q1, R1, S1, M1, M11)$. Drum Layers Mode and Long Chords Mode are active. The user can also finish the piece with a P^3 sign.

1. P^3 finishes the piece, as seen in the last section.

2. P^1 makes the drum layers mode inactive and the short chords mode active.

3. P^2 causes A_3 to release a long chord.

4. REP records a new voice in the drum layers mode.

As we can see, the first two actions are meta-actions, they change the state of the meta automata $META1$ and $META2$. The third action is the use of the long chord mode and the fourth action is the use of the drum layers mode. This means in this state $(Q1, R1, S1, M1, M11)$ both the drum layers and the long chords mode are active.

Let's imagine the user chose REP and recorded a bass drum loop. Then $\Im$ will change to state $(Q2, R1, S1, M1, M11)$. Again, the user has the same four options as before. Repeatedly choosing REP will not change this situation.

At a certain point, let's say $(Q3, R1, S1, M1, M11)$ the user can decide to use the short chords mode. To do that, he or she must play phrase P^1 and get to the state $(Q3, R1, S1, M1, M12)$. As $M12$ represents A_2, the short chords mode becomes active. In this state, each attack is interpreted as an ATT sign and releases a short chord, as explained before.

Notice that the drum loop can still be heard. This is because although the user cannot record another drum voice ($M12$ does not represent A_1) the sound generated by A_1 does not stop when the user leaves this mode. This means they are transparent modes. Also, they are excludent because $META2$ cannot be in $M11$ and $M12$ simultaneously, that is: the user cannot record a drum layer while playing the short chords.

Let us say the user plays five attacks and gets to state $(Q3, R6, S1, M1, M12)$. The possible actions are:

1. P^3 to finish the piece.

2. P^1 to come back to drum layers mode.

3. P^2 to release a long chord.

4. ATT to release a short chord.

We should notice that phrases P^1, P^2 and P^3 are specific sequences of attacks which in this mode will be interpreted as ATT signs and will release short chords. This means that if the user decides to, let's say, come back to the drum layers mode using P^1, the short chords will be heard until the last attack of P^1 and then will stop.

If P^1 has four notes, the current state will be $(Q3, R10, S1, M1, M11)$. This means the transition $((Q3, R9, S1, M1, M12), ATT, (Q3, R10, S1, M1, M12))$ that changes from state $R9$ to $R10$ of A_2 came together with the transition $((Q3, R10, S1, M1, M12), P^1, (Q3, R10, S1, M1, M11))$ that changes from state $M12$ to $M11$ of $META2$.

In this situation, the player can use

1. P^3 to finish the piece.

2. P^1 to come back to short chords mode.

3. P^2 to release a long chord.

4. REP to record an hi-hat loop.

Let us say he or she decides to play a P^2 phrase. The transition generated by this action is reflexive, that means $\Im$ leaves and enters the same state $(Q3, R10, S1, M1, M11)$. This is because A_3 describes a reflexive arch $(S1, P^2, S1)$, or in this case $\Im$ describes

$$((Q3, R10, S1, M1, M11), P^2, (Q3, R10, S1, M1, M11)).$$

Although $\Im$ is in the same state as before there has been an auditive result of this transition: all the audio has been stoped for 2 seconds while a long chord has been played.

Finally, let's say the user decides to finish the piece. He or she plays phrase P^3 and $\Im$ goes to the final state $(Q3, R10, S1, M2, M12)$.

I give a whole summary of actions and transitions of the Automata Tree $(\Im, <)$ and in this way I explicitly define the sets Σ and δ omitted until now:

Actions

Actions of Automata Tree of Drum-Machine Case Study	
action	sign
REP	repetition of rhythmic phrase
ATT	attack detection
P^n	detection of phrase n

Transitions

Transitions of Automata Tree of Drum-Machine Case Study			
n^o	initial state	action	end state
1	$(Q1, R1, S1, SET(n), M11)$	REP	$(Q1, R1, S1, SET(n+1), M11)$
2	$(Q1, R1, S1, SET3, M11)$	REP	$(Q1, R1, S1, M1, M11)$
3	$(q, r, s, M1, M11)$	P^1	$(q, r, s, M1, M12)$
4	$(q, r, s, M1, M12)$	P^1	$(q, r, s, M1, M11)$
5	$(Q(n), r, s, M1, M11)$	REP	$(Q(n+1 mod 4), r, s, M1, M11)$
6	$(q, R(n), s, M1, M12)$	ATT	$(q, R(n+1 mod 12), s, M1, M12)$
7	$(q, r, S1, M1, m)$	P^2	$(q, r, S1, M1, m)$
8	$(q, r, s, M1, m)$	P^3	$(q, r, s, M2, m)$

7.1.2 Musical Examples

As I let clear in the introduction section, my final interest is to be able to create entire pieces of music using a performance-driven system. Here I present two pieces to pandeiro and computer, performed using the system described in my case study. This proves my solution solves the main addressed in this thesis.

In this section I will present both pieces and discuss the main goals and drawbacks of my approach.

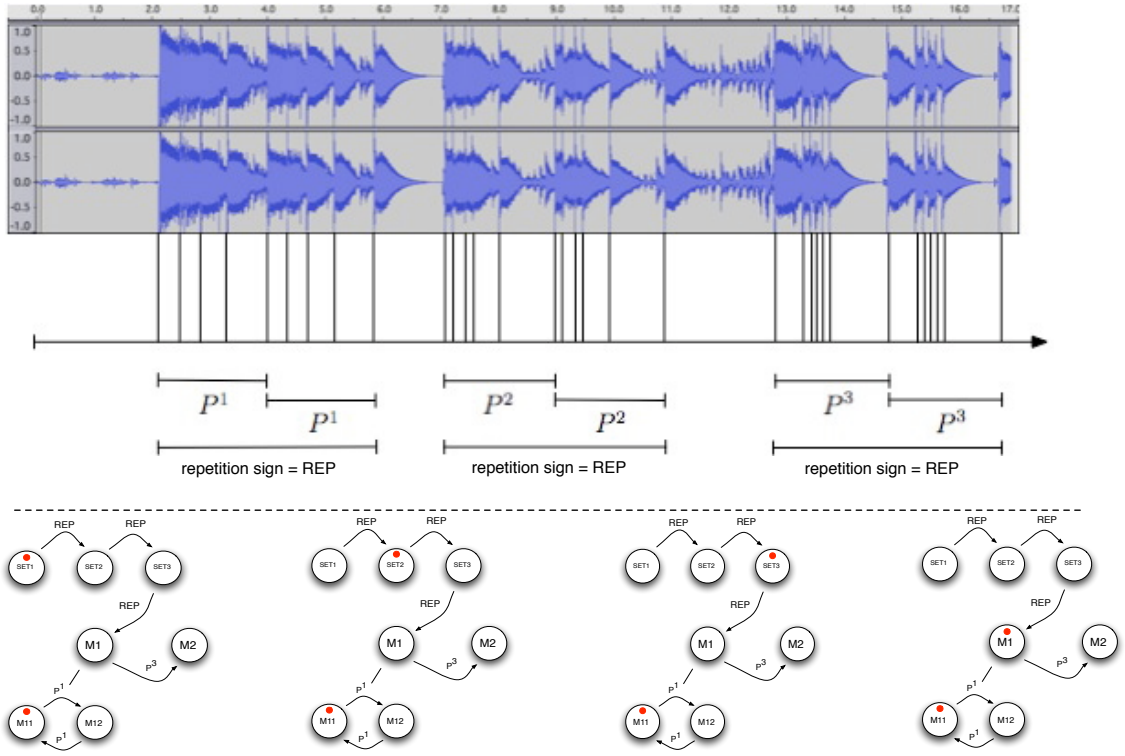

Figure 7.9: Above the dashed lines we see the audio wave and the attack signs grouped in phrase P^1, P^2 and P^3 respectively. The repetition of each phrase is the sign associated to action REP. Below the dashed line we see the state transition of the meta-automaton caused by this action each time the repetition sign is detected.

Laboratory Piece

I developed this piece at the Sony Computer Science Laboratory in Paris during an afternoon. All the section have been recorded in video. The piece had not been composed before this section, instead, it has been built up during the many trials I did while playing with the system previously described. Here I will describe only the part of this section where the piece is already settled. It can be seen at [Laboratory Piece].

Before describing the piece it's important to present the setup phase I had to go through as explained before. I extracted the audio of this phase and depicted in figure 7.9. We can see the three phrases being recorded and the state transitions of the meta-automaton that controls this recording.

From this point, I could start to use the system as mentioned above.

I depict a graphical representation of the piece in figure 7.10. In the top of this figure, we see the audio wave extracted from the video [Laboratory Piece]. Below that we see a timeline representation of the piece that shows when each sign has been detected and caused the automaton $\Im$ to change its state. Finally we see a schematic score representation where the piece is divided in seven parts. In this last representation, we see when each "voice" can be heard during the

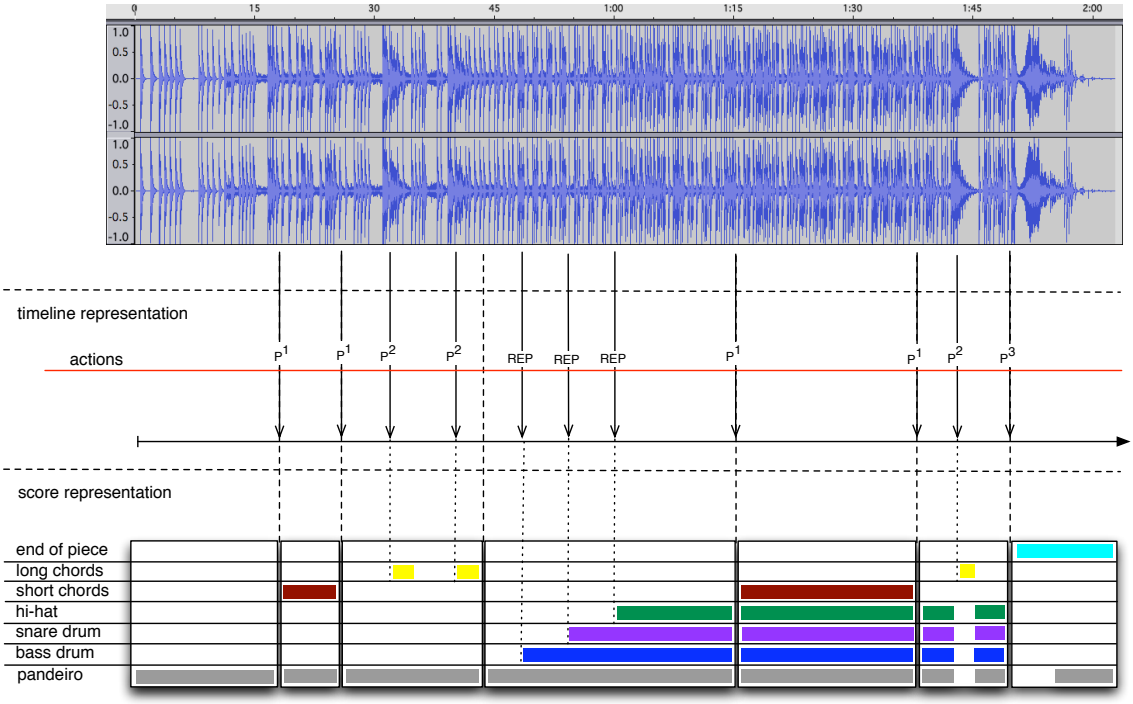

Figure 7.10: We see three representations in this picture. The audio wave, the timeline representation showing the instant each action is applied and the score representation showing a scheme of each of the seven parts of the piece.

piece. By "voice" I mean all kinds of sounds heard in the audio wave: the pandeiro played by the musician during the piece, the drum sounds generated by the computer (bass drum, snare drum and hi-hat), the harmonic sounds also generated by the computer using the continuator (short chords and long chords) and finally the group of sounds that signal the end of the piece (some chords followed by a cymbal sound).

Now, I can describe in details what happens during the piece. In the beginning, the automaton $\Im$ is in the state $(Q1, R1, S1, M1, M11)$ (notice the setup phase is finished and the automaton is already in the interactive phase). The first part of the piece is a pandeiro solo. In the end of this part, I plays phrase P^1 which is detected and interpreted as action P^1. This causes $\Im$ to change to state $(Q1, R1, S1, M1, M12)$.

Action P^1 marks the beginning of part two. As we know, in this state $(Q1, R1, S1, M1, M12)$ each attack is interpreted as action ATT (which releases a short chord) but it's important to notice these actions were omitted in figure 7.10 for the sake of visual clarity. At this point, I play twelve attacks completing the whole melodic cycle mentioned before. At each attack, $\Im$ goes from $(Q1, R1, S1, M1, M12)$ to $(Q1, R2, S1, M1, M12)$, then to $(Q1, R3, S1, M1, M12)$ and so on until reaches the state $(Q1, R12, S1, M1, M12)$ and back to $(Q1, R1, S1, M1, M12)$ again. The last four attacks form phrase P^1 which is interpreted as action P^1 again. This action takes $\Im$ back to state $(Q1, R1, S1, M1, M11)$ and marks the beginning of part three.

In this part, I improvise a bit more and play phrase P^2 which is interpreted as action P^2. This causes $\Im$ to describe an arc transition from $(Q1, R1, S1, M1, M11)$ to itself which releases a long chord. Again I improvise a bit more and plays P^2 releasing another long chord. This marks the end of part three.

In part four I repeat twice a certain phrase and this is interpreted as action REP. The automaton $\Im$ builds the bass drum voice as a looped copy of this phrase and goes to state $(Q2, R1, S1, M1, M11)$. The bass drum voice can now be heard and over that, I repeat another phrase which becomes the snare drum voice. Both voices are being played by $\Im$ which is now in state $(Q3, R1, S1, M1, M11)$. Then the I repeat the last phrase that becomes the hi-hat voice. At this moment, $\Im$ is in state $(Q4, R1, S1, M1, M11)$ and these three voices form the drum machine groove. I improvise over this groove and finish this part by playing phrase P^1.

We notice, until this moment, the piece increased in terms of musical complexity: the beginning is a pandeiro solo followed by some chords and the progressive construction of a drum machine loop. In part five ($\Im$ is at state $(Q4, R1, S1, M1, M12)$) the piece reaches the peak of its complexity because it it the moment all the drum layers are on and I expose the twelve-note melody three times. At each exposition I use a different rhythmic placement of each note creating rhythmic variations of the melody. The last four attacks of this part form phrase P^1 which turns off the chord release mode ($\Im$ goes to state $(Q4, R1, S1, M1, M11)$) and marks the beginning of part six.

In this part the drum machine groove is still on and I improvise again over this base. It is important to notice I cannot repeat another phrase unless I want the computer to record another drum machine voice (the iron triangle voice). This voice has not been recorded in this piece. During this improvisation, I played phrase P^2 causing the release of a long chord. This is an important instant of the piece because this chord release muted the whole drum machine

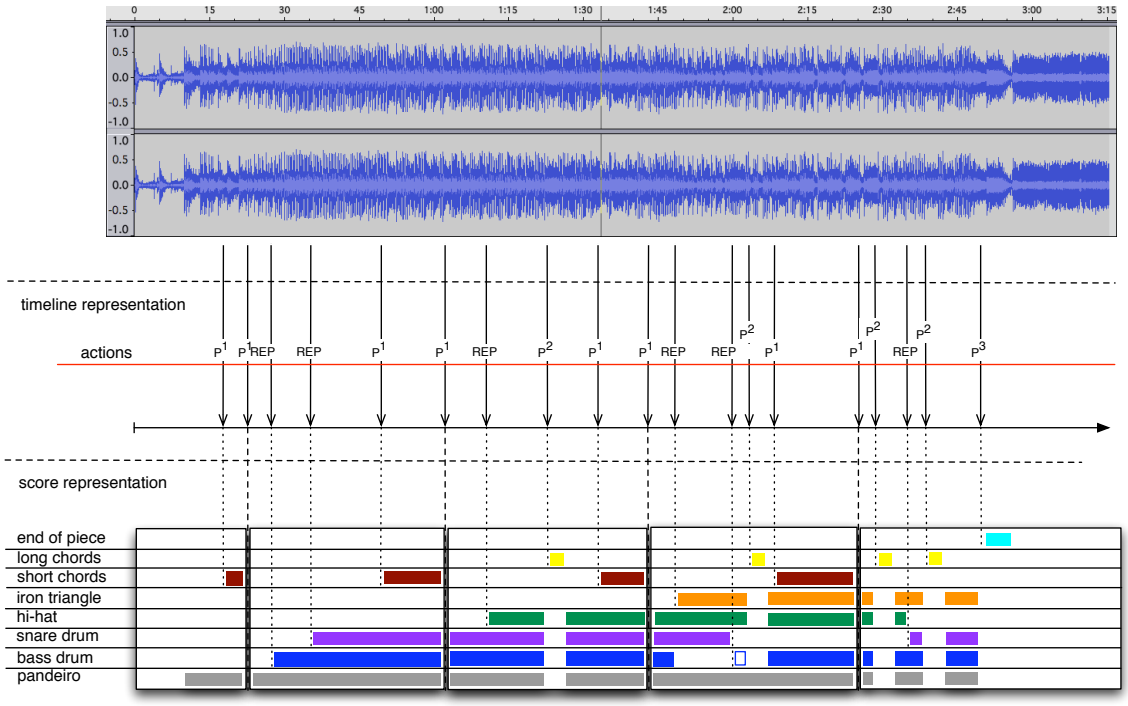

Figure 7.11: The same graphical representation is applied to this piece.

loop (this can be seen in the score representation as a gap in the three drum voices). As I also stayed in silence during the two seconds this chord sounded, a musical tension has been created. This tension is solved with the return of the drum loop and the pandeiro sounds.

Finally, to end the piece, I play phrase P^3. This makes the system mute all the audio output it was generating (in this case the three drum voices being looped) and produce a predefined sequence of sounds that end the piece. This sequence consists of three chords followed by a cymbal sound. For the sake of clarity I defined part seven as the release of this sequence together with the musical answer given by the pandeiro player which is a typical "fade out" end. In this part, the automaton $\Im$ is at the final state $(Q4, R1, S1, M2, M11)$ and the interaction is already finished because there is no possible action defined in this state.

Live Piece

The other piece presented here has been performed live at the Oldham Percussion Festival in march 2007 [Live Piece]. The system used to this performance is slightly different from the one presented before. In this case the setup phase has not been recorded on video and the phrases associated to the automata actions are not the same. The other difference is the short chord mode which uses the chromatic strategy to send the notes to the *Continuator* as I described before.

The piece is represented in figure 7.11 as before. I omit its whole description but I point out some important details that happened here.

In this case, the piece is divided in five parts. It is clear from the recording that, differently from the first one, this piece has been built in an improvised way (graphically we can see this piece is less "organized" than the first one). I had in mind a background structure: the alternation between the drum machine mode and the short chord mode, but some choices were made on stage during the performance, such as when to release long chords, the content of each drum machine voice and when to finish the piece.

As we can see in figure 7.11 each part (except the last one) finishes when I leave the short chord mode represented as red boxes in the score representation. This was the criteria used to partitioning the piece.

In general terms, this piece bear some resemblance to the first one. As we see graphically, the musical complexity also increase from the beginning to reach, at part four, its peak.

In this case, more drum machine voices were built. In the score representation we see the first two drum voices were created in the second part (bass drum represented as a blue box and snare drum represented as a magenta box), then one more (hi-hat represented as a green box) is created in the third part. The iron triangle voice, represented as an orange box, is created in the fourth part. Notice when this voice is created, the bass drum voice is stopped. As I wrote before, when the drum machine goes from state $Q4$ to $Q1$ the bass drum voice is erased and this automaton is prepared to record it once again.

In part four the bass drum voice is recorded again and the snare drum stops, but something unpredicted happened at that moment. If we look at the score representation we see two actions REP were taken in this part. The bass drum is recorded in the second REP action but right after this recording the user accidentally plays phrase P^2 which causes the release of a long chord (yellow box). As I saw before, the long chord mode is not transparent and is over the other modes, so the audio from the drum machine is muted, and we can realize the bass drum has been recorded only after the long chord stopped sounding. That is why I represented the beginning of this bass drum voice as an unfilled blue box. Finally, another snare drum voice is recorded in part five.

The short chord mode also appeared more often in this piece. As we saw before, here we do not have a melodic sense in the notes sent to the *Continuator*, but we can listen to the harmonic coherence this system gives to the sequence of chords. The first action P^1 leading to the short chords mode was not intentional. The first part of the piece was supposed to be a pandeiro solo where the player was supposed to do an *accelerando* to reach the tempo of the piece, but, during that, he played phrase P^1 which led him to the short chords mode. As that was not intentional, I soon left this mode to start recording the drum voices (part two). It is possible to notice I smiled when that happened, because it was unexpected.

The P^2 actions were concentrated in the end of the piece. As we saw before, the second P^2 action was unexpected, and I did not stop playing during the long chord sound, but the other three P^2 actions were followed by a pandeiro silence which created tension as in the first piece. The end action P^3 has been correctly interpreted.

The unpredicted detection of those two actions (the first P^1 of the piece and the second REP of part four) did not compromise the performance and we can tell that by the intensity of the applause.

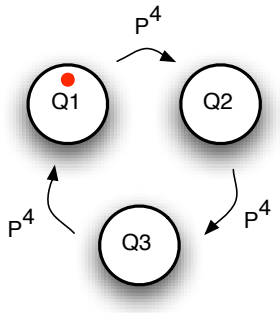

Figure 7.12: Each state represents a pre-produced loop. Every time the loop switching phrase is detected the computer changes the pre-recorded loop sample that is being played by the computer.

7.2 Pandeiro Funk Case Study

The next case study I present has some similarities with the above mentioned one. Its main goal is to allow the musician to have access to DJ tools controlled just by the rhythmic content of the musician's audio signal. The tools implemented here are a loop mode, a sampler mode and an effect mode that will be described in details.

I chose to apply these interaction tools in the aesthetic of a Brazilian genre called "Funk Carioca". In this kind of music, the rhythmic base is the central issue, usually engaging people to dance. Many DJs specialized in this style use what is called an MPC Drum Machine, a multi-pad sampler where the DJ can act almost as a percussion player. For these reasons, this music seemed to be suitable for this rhythmic interactive experience.

Adopting the same method as in the previous section, first I present the system architecture with a rigorous definition of each mode of interaction. Then I show the graphical representation of two musical pieces built using this system which were recorded on video and are available on the web ([Pandeiro Funk 1, Pandeiro Funk 2]). Finally I compare these pieces between them and to the previous ones.

7.2.1 System Architecture

As was defined before, the whole system was built using the Pure Date framework and the specific rhythm interaction tasks needed were implemented in C language as *externals*.

Each mode is now rigorously described.

Loop Mode

We can see this mode as a parallel of the drum layers mode from the Drum-Machine case study. The difference between them is that, here, the user cannot build each drum voice separately during the performance. Instead, he or she can switch between three different pre-recorded and pre-mixed loops.

In figure 7.12 we see the automaton that represents this mode. I will use the same letters to notate the automata of this case study. I call this the automaton $A_1 = (Q^1, \Sigma^1, \delta^1, i_0^1)$ where the three states are $\{Q1, Q2, Q3\} = Q^1$. The only action of this automaton is $\{P^4\} = \Sigma^1$ and it is associated to the detection of phrase P^4.

This mode is different from all the modes presented until now because it always generates audio since the moment it is turned on. In the other cases, an action was necessary to generate audio even if the mode was on: in the case of the drum-machine a repeated phrase was the command to start the audio, in the case of the short-chords mode the attacks generated each chord release. Now it is different, if this mode is on, one of the three looped samples is being played. The user has the option of switching from one loop to the other.

In its initial state the automaton plays the first loop. If P^4 is detected, it is interpreted as action P^4. The automaton changes to the next state and switches to the next looped sample. When it reaches state $Q3$ the next state will be $Q1$ back again.

The focus of this interaction is not the building of each drum layer as was before in the drum-machine mode. In this case, the player looses the freedom of building, on stage, the rhythmic structure of the loop, but wins the advantages of using pre-recorded and pre-mixed looped samples. I can list four of these advantages: pre-mixed loops usually sound better than those built live; there is no risk of mistakes while building the loop as there were before; the loops can be related to a cultural background well known by the audience which facilitates their engagement in the embodied experience of dance; and as the number of pre-recorded looped samples can be as many as the user wants, this can compensate for the lack of freedom of not being able to create the loops live.

The tempo of these loops is controlled very easily as we will see in the description of the meta-mode.

Sample Mode

Again we can see this mode as a parallel of the Mode 2 described above. Here the short chords will be substituted by small (around 200 ms) pre-recorded samples.

The automaton that represents this mode is $A_2 = \{(Q^2, \Sigma^2, \delta^2, i0^2)\}$ and is depicted in 7.13. The four states $\{R1, R2, R3, R4\} = Q^2$ accept the two reflexive actions PA, $TUNG$ and the action P^5 that changes states. PA is associated to the 'pa' detection sign, $TUNG$ to the 'tung' and P^5 is associated to phrase P^5.

Each of the four states represent a pair of two pre-recorded sounds commonly found in the "Funk Carioca" style. As said before, the classification of 'pa' and 'tung' sounds has been implemented, and this allows the use of two different attack signs. When this mode is active, each attack of the user is classified as a 'pa' or a 'tung' and each one releases one of the two sounds of the pair of the current state.

The 'pa' attack sign is associated to the PA action and the 'tung' to $TUNG$. When the user sends a sign, the system generates the respective sound and comes back to the same state, waiting for the next attack sign to be classified.

When the user wants to change to the next pair of sounds, it suffices for him or her to play

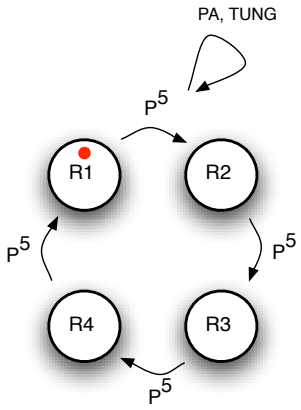

Figure 7.13: Each state represents a pair of samples associated to the 'pa' and 'tung' detection signs respectively. To change this pair, phrase P^3 must be played

phrase P^5 whose detection is associated to action P^5. As seen in the picture, after state $R4$ the automaton comes back to state $R1$.

The difference between this mode and the short chords mode previously presented is that I do not use the Continuator to generate interesting short chords. My interest here is on dealing with percussive phrases created with pre-recorded sampled sounds in the same sense the DJs do using their MPC. These two-sound phrases can be rhythmically very complex as we will see in the musical examples.

As said before this system has been used in a project that focus on the "Funk Carioca" aesthetics, but the use of sounds "attached" to the musicians notes can be useful in many other cultural contexts. In fact, we can tell that by the number of pedal-like systems found in the market for this purpose. As far as I know, none of them is controlled uniquely by the sound.

Effect Mode

This mode has been implemented recently and unfortunately is not used in the musical examples I present. It has been used in the SIGGRAPH presentation ([Krakowski 09]) but that solo had not been recorded on video. Nevertheless I believe it worth describing how this mode works.

The motivation for building it comes from the fact that very simple effects applied over a small number of sounds can generate a very large number of different sonic results.

By effect, I mean digital filtering processes applied to offline or online recorded sounds.

There is an essential difference from this mode to all the others presented until now: it works only if there is already some sound being generated. The filters act on this sound distorting up to a point the original sound can become unrecognizable.

As we see in the picture 7.14, $A_3 = \{(Q^3, \Sigma^3, \delta^3, i0^3)\}$ that represents the effect mode has three states $\{S1, S2, S3\} = Q^3$ and all of them accept the reflexive action ATT. The action P^6 changes states.

In the previously presented modes of interaction, the sonic result of each state transition (reflexive or not) did not differ much from one to the other. The sample mode changes samples

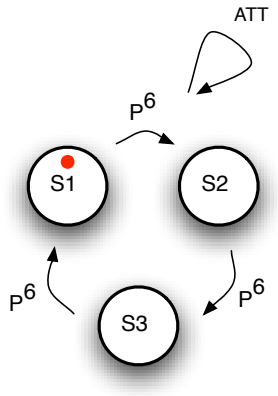

Figure 7.14: Each state represents an effect that will filter the sounds already being generated by the machine. All the effects use the attack sign as a parameter to the filtering. This is represented by the reflexive arrow in the corner of the picture.

in each state, but the PA and $TUNG$ sign always releases instant sampled sounds, when the loop mode changes states, the looped audio sample changes, but there is still a loop going on.

As the sonic result of the filters is so different, I could model each effect as a mode of interaction, but as all of them have only one intrinsic action, namely ATT, and for the sake of clarity, I decided to model them as one automaton. I describe in details how each one of them works.

Sound convolution The first state $S1$ represents the sound convolution effect. It works in the following way: every time an attack sign ATT is generated, the content of the audio of the pandeiro attack is convoluted to the audio generated by the computer. This is done by multiplying the windowed FFT result of this audio with the spectrum of that attack.

This makes the audio generated by the computer depend on the timbral characteristics of the last played attack.

Delay In this state, each attack turns on and off a delay line applied to the audio generated by the computer. This creates interesting rhythmic effects since the percussive content of the computer audio is repeated in a delay fashion.

Distortion Finally, in this mode, the audio from the computer is filtered by a typical distortion effect. This mode is useful for creating dynamic situations since the distortion is turned on and off at each attack of the performer.

Long Sample Mode

The long chord mode present in my first case study has an analog in this system but I omit it because of its simplicity and to avoid repetition. When its result appears in the examples, I comment it apart.

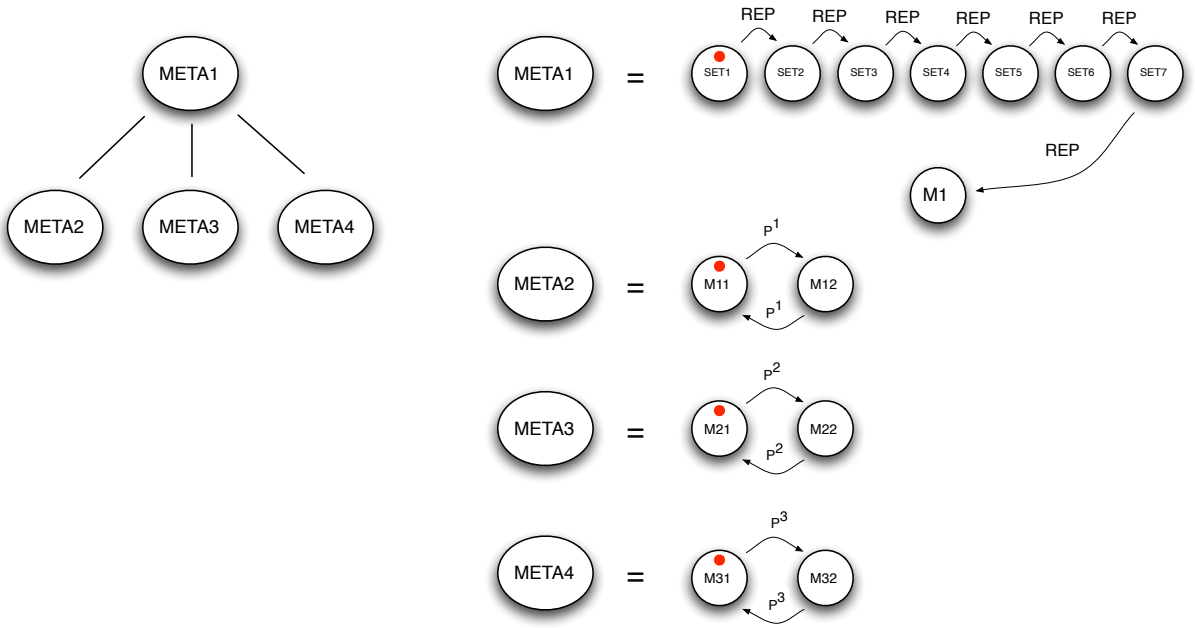

Figure 7.15: Tree structure of the meta-mode automaton that controls the system. Each meta-automaton that forms the meta-mode is graphically defined on the right of this picture.

Meta-mode

In figure 7.15 we see the tree structure of the meta-mode that controls the activity of the modes of interaction. In this case study this structure is a bit more complex than the last one: $META2$, $META3$ and $META4$ have $META1$ as the parent node.

In figure 7.16 I depict the whole meta-mode. As we can see $M1$ represents all the other meta-automata.

The setup phase is much longer than the previous case study, from $SET1$ to $SET7$ but the recording method is the same, each phrase $P^1, P^2, ..., P^7$ must be repeated. They will be used as signs to be associated to the actions of the automaton $\Im$ that models the system.

Again the interactive phase is represented by $M1$. Notice there is not a final state as was the case of $M2$ in the last case study. In this case, the piece will end when the musician stops playing and the machine stops generating audio.

The automata $META2$, $META3$ and $META4$ are the parents of A_1, A_2 and A_3 respectively. They serve as switches for these modes of interaction. When $META2$ is in state $M12$, the loop mode is on, when it is in the $M11$ state, the mode is off. When $META3$ is in state $M22$, the sample mode is on, when it is in $M21$, the mode is off. Finally, when $META4$ is in $M32$, the effects are on, when it is in $M31$, they are off. In other words, these modes are all orthogonal to each other.

The only transition that generates sound in the meta-mode is when the loop-mode is switched on: $(M11, P^1, M12)$. In fact, phrase P^1 carries the tempo of the loop. When the user plays it, the system generates a loop synchronized to the tempo in which this phrase has been played.

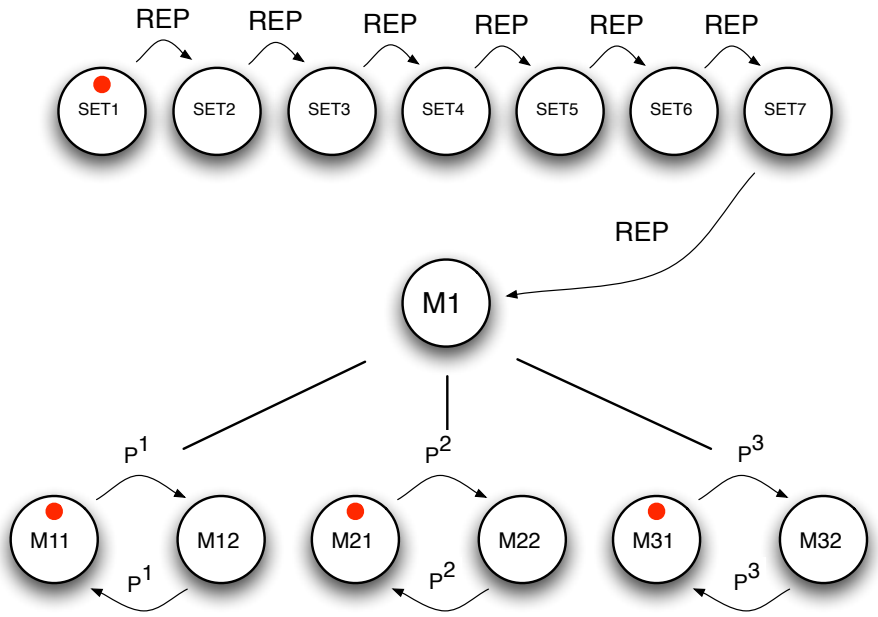

Figure 7.16: Complete meta-mode automaton $M1 = r(META2) = r(META3) = r(META4)$.

The loop sample used depends on the state of A_1. This will be clear from the examples.

The meta-mode is defined as the synchronized automaton $M = (Q^M, \Sigma^M, \delta^M, i0^M)$ over the collection $\{META1, META2, META3, META4\}$.

$$
\begin{aligned}
&META1 = \{SET1, SET2, SET3, SET4, SET5, SET6, SET7, M1\}\times \\
&\{REP\} \times \delta^{META1} \times \{SET1\} \\
&META2 = \{M11, M12\} \times \{P^1\} \times \delta^{META2} \times \{M11\} \\
&META3 = \{M21, M22\} \times \{P^2\} \times \delta^{META3} \times \{M21\} \\
&META4 = \{M31, M32\} \times \{P^3\} \times \delta^{META4} \times \{M31\}
\end{aligned}
$$

where δ^{METAi} is defined in picture 7.16. A rigorous definition will be given further on.

In other words,

$$
\begin{aligned}
&M = (Q^{META1} \times Q^{META2} \times Q^{META3} \times Q^{META4}, \{REP, P^1, P^2, P^3\}, \\
&\delta^M, (SET1, M11, M21, M31)),
\end{aligned}
$$

where again δ^M can be extracted from 7.5.

The Automata Tree

Theorem 3 *Let $T = \{A_1, A_2, A_3, META1, META2, META3, META4\}$ be the collection of automata defined in this case study.*

Let $<$ be a partial order in this set defined by

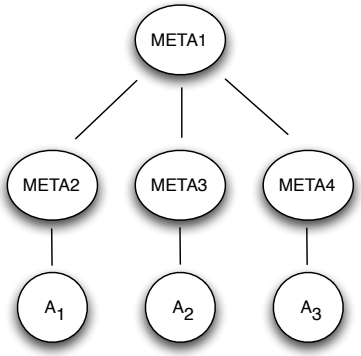

Figure 7.17: Tree structure $(T, <)$ of the whole system. Above the doted line we see the meta-mode automaton M, below it we see the modes of interaction

$$
\begin{aligned}
&META1 < META2,\\
&META1 < META3,\\
&META1 < META4,\\
&META2 < A_1,\\
&META3 < A_2, \ \textit{and}\\
&META4 < A_3.
\end{aligned}
$$

Let $\Im = (Q, \Sigma, \delta, i0)$ *be the synchronized automaton over the collection* T.
Let $r : T \to \cup\{Q_1, Q_2, Q_3, Q^{META1}, Q^{META2}, Q^{META3}, Q^{META4}, \{\emptyset\}\}$ *be defined by*

$$
\begin{aligned}
&r(A_1) = M12,\\
&r(A_2) = M22,\\
&r(A_3) = M32,\\
&r(META2) = M1,\\
&r(META3) = M1,\\
&r(META4) = M1 \ \textit{and}\\
&r(META1) = \{\emptyset\}.
\end{aligned}
$$

Then, $(\Im, <)$ *is an Automata Tree.*

Proof 2 *It suffices to show that items i and ii of 6 are satisfied.*
Item i: in fact, $\forall i \in [3]$, $\forall A \in T$, *if* A *is comparable to* $A_i \Rightarrow A < A_i$.
Item ii: given $A, B \in T$, *if they are adjacent,* $r(B) = q$ *where* q *is a state of* A.

We can see the tree $(T, <)$ depicted in 7.17.

As before, I need to force $\Im$ to satisfy the representation constraint. Done so, we can interpret figure 7.18 in the same way I did in the last case study.

I give here a more simplistic explanation on how to understand the picture. The red dots define the current state of $\Im$. A red dot of an automaton $A \in T$ can move only if its parent

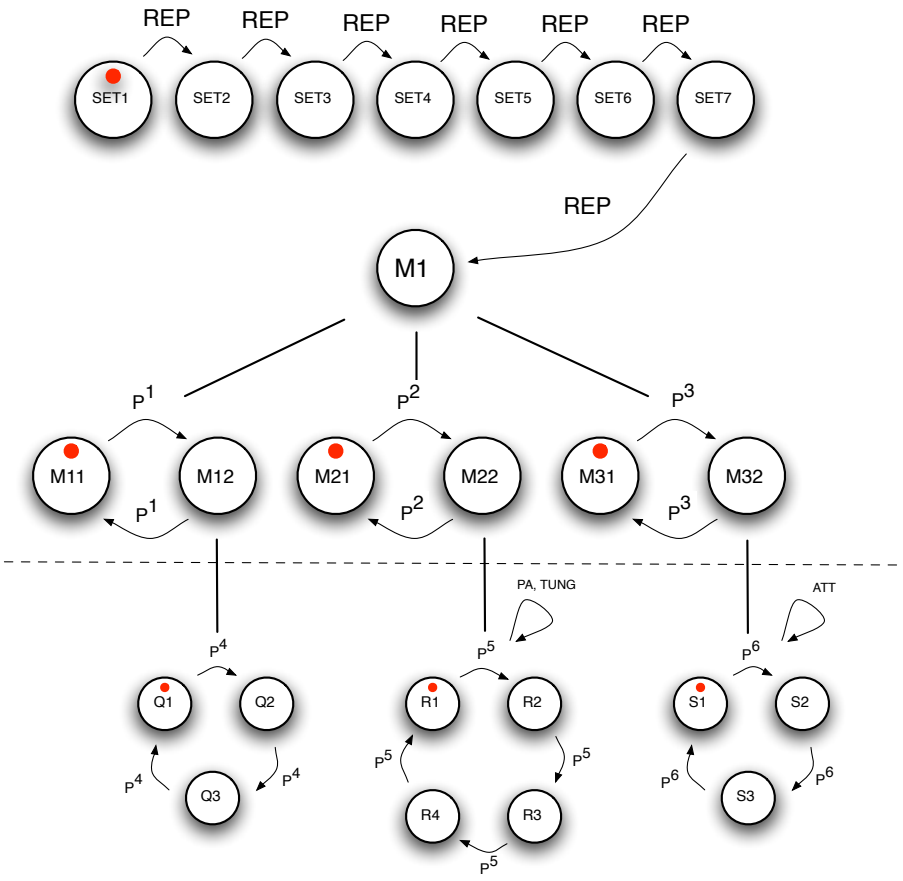

Figure 7.18: Automata Tree $(\Im, <)$ that models the system of this case study.

$P < A$ is on the state $q = r(A)$ which is linked by an edge of the tree to the automaton A. In the case of the red dot of the root automaton, this is trivially satisfied.

This Automata Tree has a longer setup phase and all the interactive phase happens when $META1$ is in state $M1$. As noticed before, there is no end state in $META1$ so the user has to finish the piece by turning off all the audio generated by the system and by stop playing.

Also as mentioned above, each mode of interaction (A_1, A_2, A_3) has a "switch" $(META2, META3, META4$ respectively) and they are active only if the "switch" is in the right state. Their activity is independent of the others', or in other words, they are orthogonal.

All that can be extracted from figure 7.18. About the interaction between the audio from each mode, we shall notice the loop and the sample modes are transparent and the effect mode, by definition, filters both of them. The long sample mode omitted in this representation is also transparent.

Finally I shall notice that only six signs (up to P^6) have been used but the setup phase recorded seven phrases. P^7 is used to release the long sample sound that has been omitted in the description or each mode. When this phrase is used in the examples I treat it apart as said before.

In this case I skip the hypothetical example to go to the real musical experiences, but before that I present the action and transition tables of this case study.

Actions

Actions of Automata Tree of the Pandeiro Funk Case Study	
action	sign
REP	repetition of rhythmic phrase
ATT	attack detection
PA	'pa' sound detection
TUNG	'tung' sound detection
P^n	detection of phrase n

Transitions

Transitions of Automata Tree of the Pandeiro Funk Case Study			
n^o	initial state	action	end state
1	$(Q1, R1, S1, SET(n), M11, M21, M31)$	REP	$(Q1, R1, S1, SET(n+1), M11, M21, M31)$
2	$(Q1, R1, S1, SET3, M11, M21, M31)$	REP	$(Q1, R1, S1, M1, M11, M21, M31)$
3	$(q, r, s, M1, M11, m2, m3)$	P^1	$(q, r, s, M1, M12, m2, m3)$
4	$(q, r, s, M1, M12, m2, m3)$	P^1	$(q, r, s, M1, M11, m2, m3)$
5	$(q, r, s, M1, m1, M21, m3)$	P^2	$(q, r, s, M1, m1, M22, m3)$
6	$(q, r, s, M1, m1, M22, m3)$	P^2	$(q, r, s, M1, m1, M21, m3)$
7	$(q, r, s, M1, m1, m2, M31)$	P^3	$(q, r, s, M1, m1, m2, M32)$
8	$(q, r, s, M1, m1, m2, M32)$	P^3	$(q, r, s, M1, m1, m2, M31)$
9	$(Q(n), r, s, M1, M12, m2, m3)$	P^4	$(Q(n+1 mod 3), r, s, M1, M12, m2, m3)$
10	$(q, R(n), s, M1, m1, M22, m3)$	P^5	$(q, R(n+1 mod 4), s, M1, m1, M12, m3)$
11	$(q, r, S(n), M1, m1, m2, M32)$	P^6	$(q, r, S(n+1 mod 3), M1, m1, m2, M32)$

7.2.2 Musical Examples

The reason why the "Funk Carioca" aesthetic directed the choice of the audio samples used in these modes of interaction, is because the system I present in this case study was designed to be used in a series of concerts that happened in Rio de Janeiro during the month of January 2009 in a project called ChoroFunk.

This project was idealized by the author of this thesis in an attempt to mix together two different styles of music, the already mentioned Funk Carioca and the Choro. The last one is known to be a traditional popular genre that dates back to the mid-XIXth century and that now is seen as a symbol of virtuosity. On the other side, the Funk is seen as mass phenomenon by the elite, something not polite at all.

Mixing together high level instrumental music and rough new Brazilian electronic music was possible only because in its root, the Rhythm, they have many common aspects.

The author of this thesis used this system to perform a pandeiro-computer improvised solo at each ChoroFunk concert.

Each solo presented there was a very difficult challenge. In real-life situations, all kinds of problems can and do happen on stage. From technical problems with the sound equipment (that place has a very bad one), to cables that disconnect during the presentation (this really happened), and microphones that break because of the heat and humidity (this also happened in one concert of this series).

For sure there is no better place to test men-machine music interaction up to its limits than that situation. All this solos of that series were recorded in video and two of them were chosen to serve as real-life musical examples of this thesis.

I now proceed to a description of each one of them.

Pandeiro Funk the 9th January 2009

This improvised solo happened in Rio de Janeiro at the Clube dos Democraticos the 9th january 2009.

As before, I omit the setup phase. In these examples the effects mode is not used as I mentioned before. Figure 7.19 shows the representation of the system without that mode. The phrases P^3 and P^6 related to it will be ignored although theoretically they have been recorded in the setup phase. This will not create any problem to analyse the piece.

We see in figure 7.20 the same musical representation applied to this piece.

We should notice here the voices of the rhythmic base do not sum up as in the other musical examples. Instead each rhythmic loop is excludent; as I pointed out before, the musician can choose what loop to use, but cannot build it live as in the previous examples. The pros and cons of this approach have already been discussed in the mode's presentation. Also, each pair of sampled sounds is also excludent. This can be seen in the picture: there is no moment where the computer generates more than two voices, one is a loop, the other is a pair of sampled sounds "attached" to the 'tung' and 'pa' signs.

As we can watch in [Pandeiro Funk 1], the piece starts with a Pandeiro solo with no interaction with the computer. Then, after about one minute since the beginning of the piece, I play

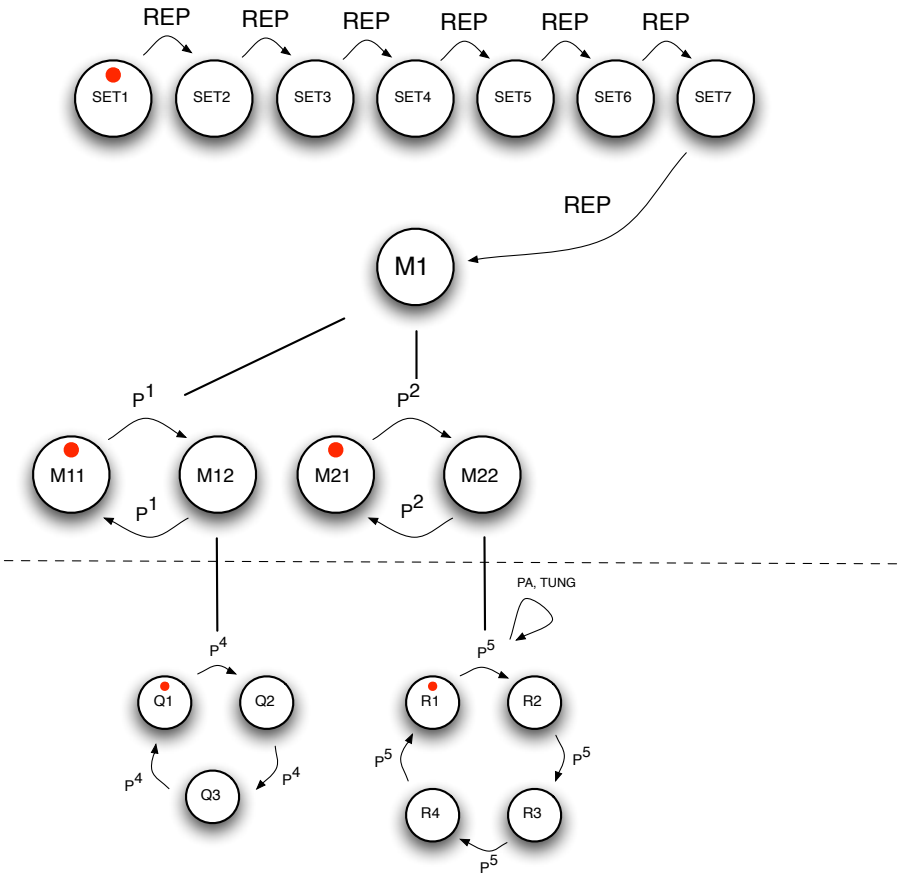

Figure 7.19: System without effects mode.

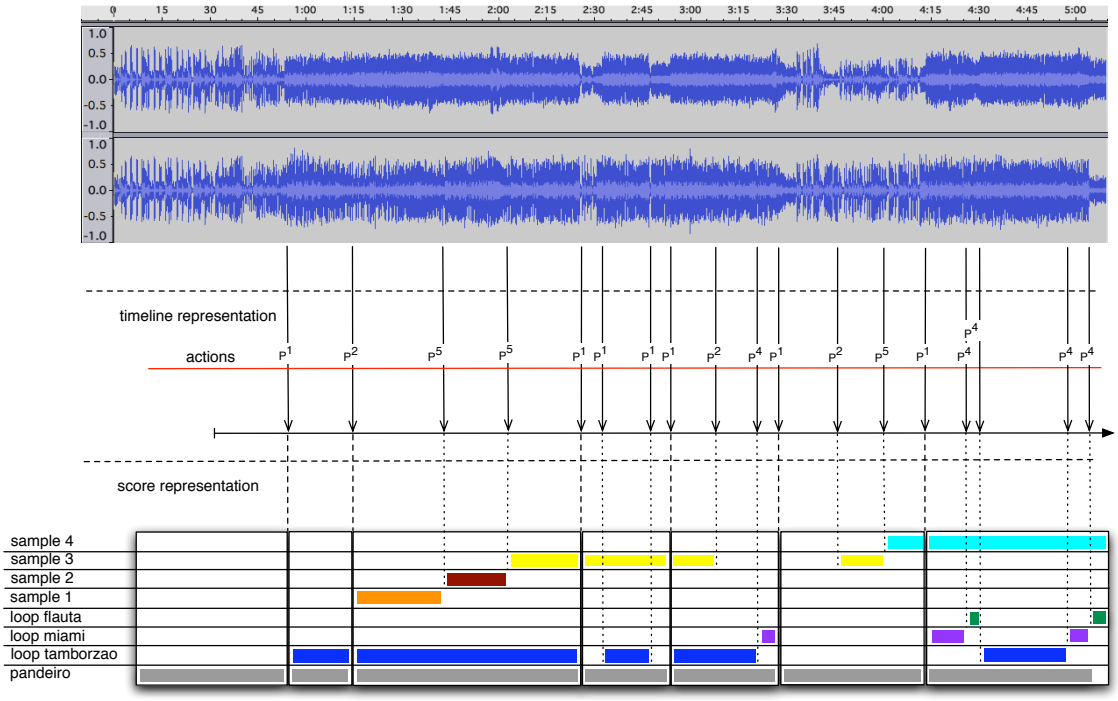

Figure 7.20: Pandeiro Funk, the 9th January 2009, live at Democraticos.

phrase P^1 and the first loop turn on (blue voice called tamborzao loop in the picture).

In this first part I improvise over that loop until around 01'10". At that moment I play phrase P^2 but the computer do not recognize it. The audience do not seem to perceive this "error", just me. I repeat that phrase and it is finally recognized, taking us to the third part of this piece.

At this moment, the first pair of sounds is on (orange strip). The 'tung' sign is associated to a low-pitched bass drum sound, and the 'pa' to a rough snare drum sound. I improvise a bit using these sounds and turn to the next pair using P^5. This pair of sounds are the B and E notes of a synthesized rough trombone timbre (red strip). These sounds are commonly used in MPCs of Funk DJs, as I mentioned before. Again, after a bit of improvisation, I turn to the next pair of sampled sounds: two synthesizer chords also used often in the Funk aesthetics (yellow strip).

Unfortunately a voice can be heard in the background. This was the microphone of the special guest which was opened during the recording.

After some seconds of more improvisation, I turn off the loop mode using phrase P^1, and pass to the fourth part of the piece as we see in picture 7.20.

At this point the piece arrives at a suspension moment. The loop flow has been taken out and I play more sparse phrases. The loop is reestablished with P^1 and taken out again, coming back to a suspension. Then I accelerate the beat of the solo and play P^1 in this new tempo.

As my system has a tempo flexibility, the computer restarts the loop synchronized to my playing, which is essential to create the effect of increasing tension. We are taken to part five of

the piece.

I continue improvising over this tempo using the same pair of samples I left on since part three. In the middle of part five I play P^2 turning off these samples and continue improvising. At 3'15", again, the computer misses a P^4 phrase I did. Again, the audience do not seem to notice it and when I retry this phrase the system understands it and changes to the Miami loop (purple).

After some repetitions of this loop, I turn off the loop mode with phrase P^1 and come back to a solo situation (part six). Although we could expect a bad response from the audience, since the piece becomes less energic without the electronic sounds, they react well. Then I choose to come back with the computer sounds gradually: first I turn on the same samples (yellow) using P^2 and switch to new samples (cyan) using P^5. Then, I restart the loop mode generating a good response from the audience (part seven).

I switch for the last loop (flauta loop in green) and come back to the tamborzao loop using P^4 twice. Finally I play P^4 twice again and finish the piece. In fact, this solo do not end because the special guest comes on stage and I play together using the green loop as a rhythmic base. As there are no other commands during that part, except a P^1 turning off the system, I left it outside my discussion.

This piece is not very well structured as we can see in the picture. The timbre choices do not change much and the middle part seems a bit confusing in my point of view. On the other side, the acceleration effect has a very good response and creates a climax point in the beginning of the seventh part.

Pandeiro Funk the 9th January 2009

This other solo took place in the same series of concerts, two weeks after the previous one. The major difference between them is that in this case I used my system linked to a VJ's computer and each pair of sampled sound is linked to a pair of pre-defined videos.

Except for this image generation, all the other details regarding the system are the same as those presented above. The consequences of using image attached to sound will be discussed here.

Figure 7.21 shows the same graphical representation used before. Again, the piece starts with a solo Pandeiro part. I play phrase P^1 and start the tamborzao loop (blue strip that starts part two). After some cycles of it, I turn it off with P^1. We can hear some people started clapping their hands. This on-off strategy can be observed in the beginning of several Funk DJ presentations. I improvise a bit more, turn the tamborzao on and off again with P^1 passing to part three.

In this part I do an exposition of all the loops. First I switch to the Miami loop (purple) using phrase P^4. At this point, phrase P^7 is accidentally detected. As I mentioned before, this phrase causes the computer to release a sampled sound with the words: "ei, como que essa tal de msica americana, hein?". I continue my exposition passing to the flauta loop (green) with phrase P^4. At this point, I turn on the sample mode with P^2 starting part four (orange strip).

Notice the camera man that is recording just realizes there is image generation going on and

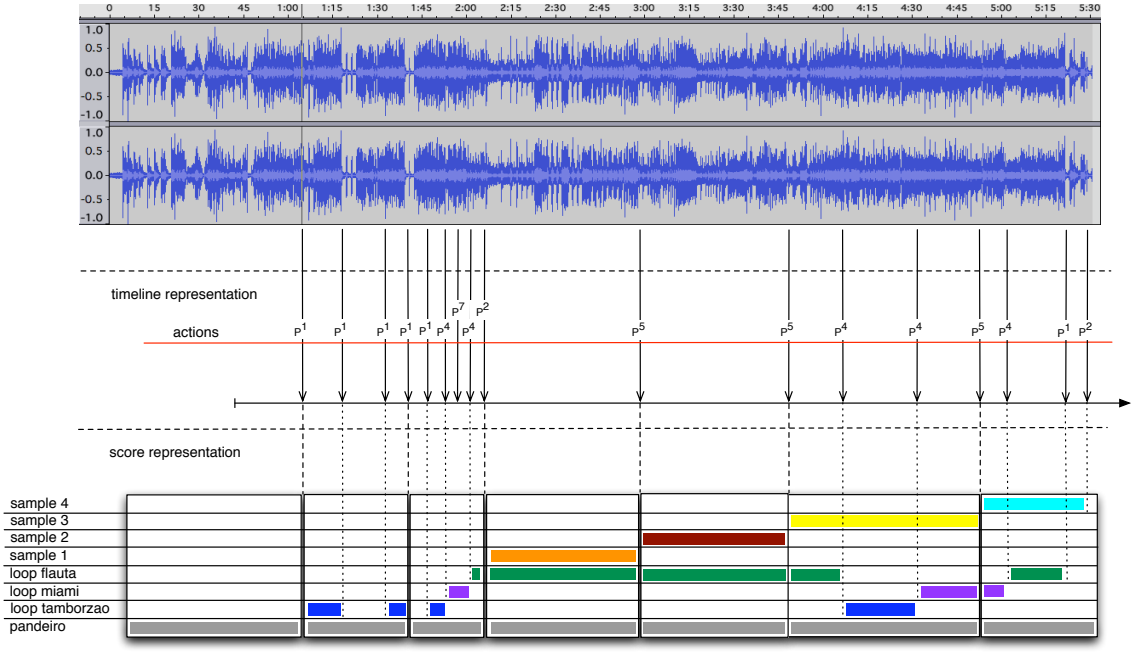

Figure 7.21: Pandeiro Funk, the 23rd January 2009, live at Democraticos.

zooms out to also capture the screen. The musical phrases I do in this part are much simpler than those of the last musical example because I am concerned about the images.

I change for the next pair of sounds/images (red) using P^5. In this fifth part something worth noticing. I started improvising freely and than, at about 3'20" I looked at the screen. Naturally, I started to "play with the image" and make phrases that let clear the connection of them with my actions. This seemed to please the audience. I continued in this way making a classical 8th-note/triolets/16th-note effect used by DJs. This "acceleration" caused a good response from the audience.

Using P^5, I changed to the next pair of sounds/images and went to the sixth part. As the flauta loop (green) have less bass sounds than the other loops, I took advantage of that and did a crescendo followed by a loop switch to the tamborzao, using phrase P^4. The audience answered positively. At that point I did not pay much attention on the images and concentrated on the musical aspect of the improvisation. Accidentally, P^4 has been detected and the Miami loop (purple) substituted the former one. I continued the improvisation until I changed to the last pair of sounds/image (cyan) starting the seventh part.

Finally, I came back once again to the flauta loop and finished the performance with a P^1 phrase that stoped that loop. Notice that after people the audience applauded, I turned off the sample mode with P^2 as it was still on, but inactive.

As I pointed out, in the fourth and fifth part, the fact of generating image forced me to play in a different way, both to "explain" that to the public, and to "play with the image".

Although this solo seems to be more structured, the audience's response was less enthusiastic, maybe because it lacked virtuosism, or the suspension technique used in the previous example.

Nevertheless, we can point out two important climax instants one in the middle of part five and the other in the second third of part six that seemed to please the audience.

7.3 Concluding Remarks

The first example of the Drum Machine Case Study, created in the laboratory, has a more organized characteristics and was the result of several trials. Each part is clearly divided and its musical complexity grows to culminate in a climax in the sixth part.

The second example of this case study, being live, has a less organized character. It is more difficult to define climax points in this piece. The appearance of the short and long chords modes as a strategy to capture the audience's attention worth noticing. As the Drum voices are recorded, the piece's complexity grows.

The first Pandeiro Funk example is completely different in terms of Form. As we noticed, in the sixth part I took out all the computer generated sounds. This subit decrease of complexity brought tension to the piece as we can perceive by the audience's response. The final growing explosion seemed to have worked better than the strategies applied in the other case study.

Finally, in the last piece, the use of the image brought another dimension to interaction. Although at a certain point this restricted my playing (until the fifth part of the piece), the result obtained after the audience gradually linked my acts with the images, were powerful. In the sixth part of the piece I could, then, improvise freely and arrive to a climax.

As we see, the difference of *Form* of each piece is obvious. Using this the Multi-Mode strategy allowed me to try, as a performer, several different sequences of modes of interaction breaking the usual Sequential-Mode paradigm of a pre-defined sequence of modes of interaction.

In this way I could try several non pre-defined performances configurations that let me investigate the effectiveness of some musical strategies such as growing complexity vs. oscillatory complexity, or virtuosity vs. simplicity.

This could not be done with pre-defined Sequential-Mode techniques, unless the performances were pre-programmed.

Chapter 8

Conclusions

In this thesis I present a *Framework* to deal with Interactive Music Systems which:

- Uses a Multi-Mode Paradigm;
- Is Rhythmically-Controlled; and
- Audio-Based.

The hypothesis of this thesis is that Multi-Mode Systems allow their users to create variable-form pieces. Also that Rhythmically-Controlled Systems allow their users to have freedom 'on stage' to decide what will be be computer's musical responses and that can only be achieved if the System is Audio-Based.

I then present two *Interactive Music Systems* as instantiations of this *Framework* and which:

- Use Automata Theory to model the concept of *Modes of Interaction*;
- Use The Synchronized Automata Theory augmented by Tree Structures to create the *Automata Trees* which model the concept of *Meta-Modes of Interaction*; and
- Use a formalized concept of *Rhythmic Phrases* to control the actions of these Automata.

Finally, I prove my hypothesis is correct by presenting four *Solo Improvised Pieces* built using the above *Systems* and which:

- Are performed by the musician 'on stage' having the complete freedom to use whatever *Mode of Interaction* he wanted to, and communicating so to the computer only through *Rhythmic Phrases* that served as *commands* and *meta-commands*; and
- Have completely different *Form* with different climax points (those live).

I do not use any aesthetic judgement to classify the quality of these pieces. Their video recordings are the proof my approach works to build practical results, which was my main concern. Nevertheless, my aesthetic choices were the main reference that guided all the steps of

this research and I tried to let this clear throughout this text. My research was built based on real-world examples and because of that my systems are robust enough to face real-life situations.

Besides this main result, some other contributions can be stated.

I make a detailed analysis about many approaches to Rhythm in an attempt to open the possibilities of future Computer Music works.

My approach is focused on musical effectiveness and not in scientific development by itself, although I believe the latter is also achieved in this thesis as a consequence of the former.

I hope my work serves also to prove it is possible to do Computer Music applied to Popular Music. My intention was to deal with this Music without loosing the complexity of virtuous improvisation which is intrinsic to it.

8.1 Future Work

As I mentioned in other parts of this text, in future works I intend:

- To apply the Pandeiro sound classification results we had in [Roy, Pachet & Krakowski 07/2] to improve the power of the low-level analysis.
- To use other features and Machine Learning techniques to improve the controllability of the *Rhythmic phrases*; and
- To apply Machine Learning techniques to improve the flexibility of the *Meta-Modes*.

Many other directions of research can be taken. I intend to try other Modes of Interaction and other sonic solutions to the actual architecture I have in the attempt of creating a more complete artistic result using my *Framework*.

Bibliography

[1] A. Klapuri, M. Davy, *Signal Processing Methods for Music Transcription*. Springer, New York, 2006.

[Roads 96] C. Roads, *The Computer Music Tutorial*. MIT Press, Cambridge, 1996.

[2] P. Cook, *Real Sound Synthesis for Interactive Applications*. A K Peters, Natick, 2002.

[3] F. Moore, *Elements of Computer Music*. Prentice Hall, New Jersey, 1990.

[Fraisse 82] P. Fraisse, Rhythm and Tempo. In D. Deutsch (Ed.) *The Psychology of Music* (pp.149-180). New York: Academic Press, 1982.

[Clarke 99] E. Clarke, Rhythm and Timing in Music. In D. Deutsch (Ed.) *The Psychology of Music* (pp.473-500). New York: Academic Press, 1999.

[Agawu 03] K. Agawu, *Representing African Music*. Routledge, New York, 2003.

[Gabrielsson 93] A. Gabrielsson, The Complexities of Rhythm. In T. Thige, W. Dowling (Ed.) *Psychology and Music: The Understanding of Melody and Rhythm* (pp. 93-120). Lawrence Erlbaum Associates, New Jersey, 1993.

[Lerdhal & Jackendoff 83] F. Lerdahl, R. Jackendoff, *A Generative Theory of Tonal Music*. The MIT Press, Cambridge, 1983.

[Cooper & Meyer 60] G. Cooper, L. Meyer, *The Rhythmic Structure of Music*. The University of Chicago Press, Chicago, 1960.

[Rowe 93] R. Rowe, *Interactive Music Systems*. The MIT Press, Cambridge, 1993.

[Rowe 01] R. Rowe, *Machine Musicianship*. The MIT Press, Cabridge, 2001.

[Pachet 02/2] F. Pachet, Interacting with a Musical Learning System: The Continuator. In *Music and Artificial Intelligence* (103-108). Springer, Berlin, 2002.

[Tanguiane 93] A. Tanguiane, *Artificial Perception and Music Recognition*. Springer-Verlag, New York, 1993.

[Smalley 86] D. Smalley, Spectro-morphology and structuring processes. In S. Emmerson (edt.), *The Language of Electroacoustic Music*(pp. 6193), The MacMillan Press Ltd, Basingstoke, England, 1986.

[Handel 89] S. Handel, *Listening: An Introduction to the Perception of Auditory Events*. MIT Press, Cambridge, England, 1989.

[Berliner 94] P. Berliner, *Thinking in jazz: the infinite art of improvisation*. University of Chicago Press, USA, 1994.

[Button 90] G. Button, Going Up a Blind Alley: Conflating Conversation Analysis and Computational Modelling. In P. Luff, N. Gilbert, D. Frohlich (edt.) *Computers and Conversation* (pp. 67-90). Academic Press Limited: London. 1990.

[Vianna 95] H. Vianna, *O misterio do Samba*. Jorge Zahar Editora, 1995.

[Bremaud 02] P. Bremaud, *Mathematical Principles of Signal Processing*. Spriger-Verlag, New York, 2002.

[Zwicker 99] E. Zwicker, H. Fastl. *Psychoacoustics: Facts and Models*. Springer-Verlag, Berlin, 2nd edition, 1999.

[London 04] J. London, *Hearing in Time: Psychological Aspects of Musical Meter*. Oxford University Press, New York, 2004.

[4] M. Puckette, *The Theory and Technique of Electronic Music*. URL: http://crca.ucsd.edu/ msp/techniques.htm. Accessed in 2009.

[5] PLOrk: The Princeton Laptop Orchestra. URL: http://plork.cs.princeton.edu. Accessed in 2009.

[Max] Cycling '74 Max 5. URL: http://www.cycling74.com/products/max5. Accessed in 2009.

[Supercollider] SuperCollider: A Real Time Audio Synthesis Programming Language. URL: http://www.audiosynth.com. Accessed in 2009.

[Chuck] ChucK: Strongly-Timed, Concurrent, and On-The-Fly Audio Programming Language. URL: http://chuck.cs.princeton.edu. Accessed in 2009.

[ReacTable] Reactable. URL: http://www.reactable.com. Accessed in 2009.

[Puredata] Pure Data. URL: http://puredata.info. Accessed in 2009.

[Haile Web] Haile Robot Homepage. URL: http://www.cc.gatech.edu/ gilwein/Haile.htm Acessed in 2009.

[Haile Pow] Powwow piece composed to Haile. URL: http://www.coa.gatech.edu/%7Egil/powwow.mov Accessed in 2009.

[Haile Jamaa1] Jam'aa piece composed to Haile and performed in Jerusalem. URL: http://www.coa.gatech.edu/%7Egil/Jam%27aaJerusalem.mov Accessed in 2009.

[Haile Jamaa2] Jam'aa piece composed to Haile. URL: http://www.coa.gatech.edu/ Accessed in 2009.

[Haile Jamaa3] Jam'aa piece composed to Haile. URL: http://www.coa.gatech.edu/ Accessed in 2009.

[Continuator] Continuator Web Site. URL: http://www.csl.sony.fr/ pachet/Continuator/ Accessed in 2009.

[Continuator Audio] Audio of Continuator Web Site. URL: http://www.csl.sony.fr/ pachet/Continuator/audio/ Accessed in 2009.

[Ircam] Score-Following web page at IRCAM. URL: http://imtr.ircam.fr/index.php/ Accessed in 2009.

[Coleman Web] S. Coleman, URL: http://www.m-base.org/sounds.html Accessed in 2009.

[Pandeiro.com] S. Feiner, Comercial Pandeiro Site. URL: www.pandeiro.com Accessed in 2009.

[Early Experiences] S. Krakowski, Early Experiences. Recorded at IMPA in 2006. URL: http://www.visgraf.impa.br/videos_tese_krakowski/pandeiro.mp4

[Aguas de Marco] S. Krakowski, Aguas de Marco. Recorded at IMPA in 2006. URL: http://www.visgraf.impa.br/videos_tese_krakowski/Aguas_100MB.mp4

[RAP] S. Krakowski, Rhythm and Poetry. Recorded at Sony/CSL-Paris in 2007. URL: http://www.visgraf.impa.br/videos_tese_krakowski/Manuel.mp4

[Cabecas] S. Krakowski, Cabecas. Recorded at Sony/CSL-Paris in 2007. URL: http://www.visgraf.impa.br/videos_tese_krakowski/Reportagem.mp4

[Cabecas] S. Krakowski, Cabecas. Recorded at Sony/CSL-Paris in 2007. URL: http://www.visgraf.impa.br/videos_tese_krakowski/Reportagem.mp4

[Laboratory Piece] S. Krakowski, Laboratory Piece. Recorded at Sony/CSL-Paris in 2007. URL: http://www.visgraf.impa.br/videos_tese_krakowski/lab_piece.mp4

[Live Piece] S. Krakowski, Live Piece. Recorded at Sony/CSL-Paris in 2007. URL: http://www.visgraf.impa.br/videos_tese_krakowski/live_piece.mov

[Pandeiro Funk 1] S. Krakowski, Pandeiro Funk Piece. Recorded at Democraticos, Rio de Janeiro, the 9th January 2009. URL: http://www.visgraf.impa.br/videos_tese_krakowski/09_01_09_Pandeiro_Funk.mov

[Pandeiro Funk 2] S. Krakowski, Pandeiro Funk Piece. Recorded at Democraticos, Rio de Janeiro, the 23rd January 2009. URL: http://www.visgraf.impa.br/videos_tese_krakowski/09_01_09_solo_demo.mov

[Band-in-a-box] P. Gannon, Band-in-a-Box. PG Music Inc., Hamilton, Ontario, 1991.

[Krumhansl 00] C. Krumhansl, Rhythm and Pitch in Music Cognition, *Psychological Bulletin*, **126** (2000), 159-179.

[Iyer 02] V. Iyer, Embodied Mind, Situated Cognition, and Expressive Microtiming in African-American Music, *Music Perception*, **19**, (2002), 387-414.

[Parncutt 94] R. Parncutt, A Perceptual Model of Pulse Salience and Metrical Accent in Musical Rhythms, *Music Perception*, **11**, (1994), 409-464.

[Zbikowski 04] L. Zbikowski, Modelling the Groove: Conceptual Structure and Popular Music, *Journal of the Royal Musical Association*, **129**, (2004), 272-297.

[Lewis 00] G. Lewis, Too Many Notes: Computers, Complexity and Culture in Voyager, *Leonardo Music Journal*, **10**, (2000), 33-39.

[Pachet 02] F. Pachet, The Continuator: Musical Interaction With Style, *Journal of New Music Research*, **31**, No. 1, (2002), **-**.

[Roads 85] C. Roads, Research in Music and Artificial Intelligence, *Computing Survey*, **17**, n. 2, (1985).

[Douglas 91] R. L. Douglas, Formalizing an African-American Aesthetic, *New Art Examiner*, **June**, (1991), 18-24.

[Wessel & Wright 00] D. Wessel, M. Wright, Problems and Prospects for Intimate Musical Control of Computers, *Computer Music Journal*, (2000), 11-22.

[Jorda 02/2] S. Jorda, Improvising with Computers: A Personal Survey (1989-2001), *Journal of New Music Research*, **31**, (2002), 1-10.

[Ryan 91] J. Ryan, Some remarks on musical instrument design at STEIM, *Contemporary Music Review*, **6(1)**, (1991), 317.

[Paradiso 99] J. Paradiso, The Brain Opera Technology: New Instruments and Gestural Sensors for Musical Interaction and Performance, *Journal of New Music Research*,**28**, (1999), 130-149.

[Andersen 09] H.K. Andersen, R. Grush, A Brief History of Time Consciousness: Historical Precursors to James and Husserl, *Journal of the History of Philosophy.* (forthcoming, spring 2009).

[Scheirer 98] E. D. Scheirer, Tempo and beat analysis of acoustic musical signals, *Journal of Acoustic Society of America*, **103(1)**, January (1998).

[Harel 87] D. Harel, Statecharts: A Visual Formalism for Complex Systems, *Science of Computer Programming*, **8**, (1987), 231-274.

[Kapur et al. 02] A. Kapur, G. Essl, P. Davidson, P. R. Cook, The Eletronic Tabla Controller, *Journal o New Music Research*, **31**, No. 1, (2002), **-**.

[Seddon 05] F. A. Seddon, Modes of communication during jazz improvisation, *British Journal of Music Education*, **22:1**, (2005), 47-61.

[Pachet 00] F. Pachet, Qu'est-ce qu'une melodie interessante?, *La Recherche*, Novembre 2000.

[Iversen 08] J. R. Iversen, A. D. Patel, K. Ohgushi, Perception of rhythmic grouping depends on auditory experience, *JASA*, **124(4)**, (2008), pp. 2263-71.

[Shmulevich et al. 01] I. Shmulevich, O. Yli-Harja, E. Coyle, D. - J. Povel, K. Lemstrom, Perceptual Issues in Music Pattern Recognition: Complexity of Rhythm and Key Finding, *Computers and the Humanities*, **35**, (2001), pp. 23–35.

[Jehan 05] T. Jehan. *Creating Music by Listening*. PhD Thesis, MIT, 2005.

[Cabral 08] G. Cabral. *Harmonisation Automatique en Temps Reel*. PhD Thesis, Paris 6, 2008.

[Smith 99] L. Smith. *A Multiresolution Time-Frequency Analysis and Interpretation of Musical Rhythm*. PhD Thesis, The University of Western Australia, 1999.

[Bilmes 93] J. Bilmes. *Timing is of the Essence: Perceptual and Computational Techniques for Representing, Learning, and Reproducing Expressive Timing in Percussive Rhythm*. Masters Thesis, MIT, 1993.

[Thom 01] B. Thom, *BoB: An Improvisational Music Companion*. PhD Thesis, Carnegie Mellon University, 2001.

[Scheirer 00] E. Scheirer. *Music-Listening Systems*. PhD Thesis, MIT, 2000.

[Sandvold 04] V. Sandvold. *Percussion Descriptors*. Master Thesis, University of Oslo, Norway, 2004.

[6] D. Murray-Rust. *Musical Acts and Musical Agents: theory, implementation and practice*. PhD Thesis, University of Edinburgh, 2007.

[Collins 06] N. Collins. *Towards Autonomous Agents for Live Computer Music: Realtime Machine Listening and Interactive Music Systems*. PhD Thesis, University of Cambridge, 2006.

[Gouyon 05] F. Gouyon. *A Computational Approach to Rhythm Description: Audio Features for the Computation of Rhythm Periodicity Functions and their use in Tempo Induction and Music Content Processing*. PhD Thesis, Universitat Pomeo Fabra, 2005.

[Beek 03] M.H. ter Beek, *Team Automata - A Formal Approach to the Modeling of Collaboration Between System Components*. Ph.D. thesis, Leiden Institute of Advanced Computer Science, Leiden University, 2003.

[Geber 97] G. Ramalho, *Construction dun agent rationnel jouant du jazz*. PhD Thesis, Universit de Paris VI, Paris, 1997.

[Murray-Rust 03] D. Murray-Rust, *VirtuaLatin - Agent Based Percussive Accompaniment*. Master Thesis, University of Edinburgh, 2003.

[Walker 94] W. F. Walker, *A Conversation-Based Framework for Musical Improvisation*. PhD Thesis, University of Illinois at Urbana-Champaign, 1994.

[Pelz-Sherman 98] M. Pelz-Sherman, *A Framework for the Analysis of Perfomer Interactions in Western Improvised Contemporary Art Music*. PhD Thesis, University of California, 1998.

[Murray-Rust 07] D. Murray-Rust, *Musical Acts and Musical Agents: theory, implementation and practice*. PhD Thesis, University of Edinburgh, 2007.

[Schweitzer 03] K. G. Schweitzer, *Afro-Cuban Bat Drum Aesthetics: Developing Individual and Group Technique, Sound, and Identity*. PhD Thesis, University of Maryland, 2003.

[FitzGerald 04] D. FitzGerald, *Automatic Drum Transcription and Source Separation*. PhD Thesis, Dublin Institute of Technology, 2004.

[Termens 04] E. G. Termens, *New Approaches for Rhythmic Description of Audio Signals*. PhD Thesis, Pompeo Fabra Universtity, 2004.

[7] M. Cooper, J. Foote, Summarizing Popular Music via Structural Similarity Analysis. In *Proc. of IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*.

[Foote & Cooper 01] J. Foote, M. Cooper, Visualizing musical structure and rhythm via self-similarity. In *Proc. of International Computer Music Conference*. La Habana, Cuba, 2001.

[Zils & Pachet 01] A. Zils, F. Pachet, Musical Mosaicing. In *Proc. of the Cost G-6 Conference on Digital Audio Effects*. Limerick, Ireland, 2001.

[PLOrk 06] D. Trueman, P. Cook, S. Smallwood, G. Wang, PLOrk: The Princeton Laptop Orchestra, Year 1. In *Proc. of the International Computer Music Conference*. New Orleans, USA, 2006.

[Pfleiderer 03] M. Pfleiderer, The Study of Rhythm in Popular Music: Approaches and Empirical Results. In *Proc. of the 5th Triennial ESCOM Conference*. Hanover, Germany, 2003.

[Paulus & Klapuri 02] J. Paulus, A. Klapuri, Measuring the Similarity of Rhythmic Patterns. In *Proc. of the 3rd ISMIR*. Paris, France, 2002.

[Gillet et al. 03] O. Gillet, G. Richard, Automatic Labelling of Tabla Signals. In *Proc. of the 4th ISMIR Conf*. Maryland, USA, 2003.

[Herrera al. 04] P. Herrera, V. Sandvold, F. Gouyon, Percussion-Related Semantic Descriptors of Music Audio Files. In *Proc. of 25th International AES Conference*. London, UK, 2004.

[Yoshii al. 03] K. Yoshii, M. Goto, H. Okuno, Automatic Drum Sound Description for Real-World Music Using Template Adaptation and Matching Methods. In *Proc. of the 4th ISMIR Conf.* Maryland, USA, 2003.

[Iyer al. 97] V. Iyer, J. Bilmes, M. Wright, D. Wessel, A Novel Representation for Rhythmic Structure. In *Proc. of the ICMC'97.* Thessaloniki, Greece, 1997.

[Cook 01] P. Cook, Principles for designing computer music controllers. In *Proc. of NIME'01.* Seattle, USA, 2001.

[Jorda 02] S. Jorda, Afasia: the Ultimate Homeric One-man-multimedia-band. In *Proc. of the NIME'02 Conference*. Dublin, Ireland, 2002.

[Weinberg al. 05] Weinberg, G., Driscoll S., and M. Parry, Musical Interactions with a Perceptual Robotic Percussionist. In *Proc. of the of IEEE International Workshop on Robot and Human Interactive Communication*, Nashville, 2005, TN 456-461.

[Weinberg & Driscoll 06] G. Weinberg, S. Driscoll, Robot-Human Interaction with an Anthropomorphic Percussionist. In *Proc. of the SIGCHI conference on Human Factors in computing systems*. Montreal, Canada, 2006.

[Weinberg al. 06] G. Weinberg, S. Driscoll, T. Thatcher, Jam'aa - A Middle Eastern Percussion Ensemble for Human and Robotic Players. In *Proc. of in ICMC'06*. New Orleans, 2006, pp. 464-467.

[Dannenberg 07] R. B. Dannenberg, New Interfaces for Popular Music Performance. In *Proc. of NIME'07.* New York, USA, 2007.

[Weinberg & Driscoll 07] G. Weinberg, S. Driscoll, The interactive robotic percussionist: new developments in form, mechanics, perception and interaction design. In *Proc. of the ACM/IEEE international conference on Human-robot interaction*, Arlington, USA, 2007, pp. 97-104.

[Weinberg al. 07] G. Weinberg, S. Driscoll, The Design of a Robotic Marimba Player Introducing Pitch into Robotic Musicianship. In *Proc. NIME'07.* New York, USA, 2007.

[Krakowski 09] S. Krakowski, Pandeiro Funk: Experiments on Rhythm-Based Interaction. In *Proc. of SIGGRAPH'09.* New Orleans, USA, 2009.

[Dannenberg 84] R. B. Dannemberg, An On-Line Algorithm for Real-Time Accompaniment. In *Proc. ICMC'84.*

[Desain & Honing 02] P. Desain, and H.J. Honing, Rhythmic stability as explanation of category size. In *Proc. of ICMPC'02.* Sidney, Autralia, 2002.

[Di Scipio 03] A. Di Scipio, Sound is The Interface. In *Proc. of the Colloquium on Musical Informatics*. Firenze, Italy, 2003.

[Ellis 97] C.A. Ellis, Team Automata for Groupware Systems. In *Proceedings of the International ACM SIGGROUP Conference on Supporting Group Work: The Integration Challenge (GROUP'97)*. Phoenix, USA, 1997, 415–424.

[Puckette 98] M. Puckette, T. Apel, D. D. Zicarelli Real-Time audio analysis tools for Pd and MSP. In *Proc. of ICMC'98*. Ann Arbor, USA, 1998.

[Kapur et al. 04] A. Kapur, P. Davidson, P. R. Cook, P. F. Driessen, W. A. Schloss, Digitizing North Indian Performance. In *Proc. ICMC'04*. Miami, USA, 2004.

[Aimi 07] R. Aimi, Percussion instruments using realtime convolution: Physical controllers. In *Proc. of NIME'07*. New York, USA, 2007.

[Cabral et al. 01] G. Cabral, I. Zanforlin, H. Santana, R. Lima, G. Ramalho, D'Accord Guitar: An Innovative Guitar Performance. In *Huitimes Journes d'Informatique Musicale (JIM'2001)*. Bourges, France, 2001.

[Vercoe 84] B. Vercoe, The Synthetic Performer in the Context of Live Performance. In*Proc. of ICMC'84*. Paris, France, 1984, 199–200.

[Aucouturier & Pachet 05] J.-J. Aucouturier and F. Pachet, Ringomatic: A real-time Interactive Drummer Using Constraint-Satisfaction and Drum Sound Descriptors. In *Proc. of ISMIR'05*. London, UK, 2005, 412-419.

[Wessel, Wright & Khan 98] D. Wessel, M. Wright, S. A. Khan, Preparation for Improvised Performance in Collaboration with a Khyal Singer In *Proc. of ICMC'98*. Ann Arbor, USA, 1998, 497–503.

[Wright & Wessel 98] M. Wright, D. Wessel, An Improvisation Environment for Generating Rhythmic Structures Based on North Indian Tal Patterns. In*Proc. of ICMC 1998*. Ann Arbor, USA, 1998.

[Walker 97] W. F. Walker, A Computer Participant in Musical Improvisation. In *Proc. of CHI'97 Eletronic Publications*. 1997.

[Franklin 01] J. A. Franklin, Multi-phase learning for jazz improvisation and interaction. In *Proc. Eighth Biennial Symposium on Arts and Technology*. New London, USA, 2001.

[Hamanaka 01] M. Hamanaka, M. Goto, N. Otsu, Learning-Based Jam Session System for a Guitar Trio. In *Proc. of ICMC'01*. Havana, Cuba, 2001, 467–470.

[FitzGerald et al. 02] D. FitzGerald, E. Coyle, B. Lawlor, Sub-Band Independent Subspace Analysis for Drum Transcription. in *Proc. of DAFx'02*. Hamburg, Germany, 2002.

[Gouyon, Pachet et al. 00] F. Gouyon, F. Pachet, O. Delerue, On The Use of Zero-Crossing Rate for an Application of Classification of Percussive Souds. In *Proc. of DAFx'00*. Verona, Italy, 2000.

[Herrera et al. 03] P. Herrera, A. Dehamel, F. Gouyon, Automatic labeling of unpitched percussion sounds. In *Proc. of the 114th AES Convention*. Amsterdam, The Netherlands, 2003.

[Zils, Pachet et al. 02] A. Zils, F. Pachet, O. Delerue, F. Gouyon, Automatic Extraction of Drum Tracks from Polyphonic Music Signals. In *Proc. of the WedelMusic'02*. Darmstadt, Germany, 2002.

[Chordia 05] P. Chordia, Segmentation and recognition of tabla strokes. In *Proc. of International Conference on Music Information Retrieval*. 2005.

[Chordia & Rae 08] P. Chordia, A. Rae, Tabla Gyan: A System for Realtime Tabla Recognition and Resynthesis. In *Proc. of ICMC'08*. Belfast, Northern Ireland, 2008.

[Roy, Pachet & Krakowski 07/1] P. Roy, F. Pachet, S. Krakowski, De l'interet des features analytiques: une etude pour la classification des sons de pandeiro. In *Actes de JIM'07*. Lyon, France, 2007.

[Roy, Pachet & Krakowski 07/2] P. Roy, F. Pachet, S. Krakowski, Improving the Classification of Percussive Sounds with Analytical Features: a Case Study. In *Proc. of ISMIR'07*. Vienna, Austria, 2007.

[Roy, Pachet & Krakowski 07/3] P. Roy, F. Pachet, S. Krakowski, Analytical Features for the Classification of Percussive Sounds: The Case of the Pandeiro. In *Proc. of DAFx'07*. Bordeaux, France 2007.

[Cabral et al. 07] G. Cabral, J.- P. Briot, S. Krakowski, L. Velho, F. Pachet, P. Roy, Some Case Studies in Automatic Descriptor Extraction. In *Proc. of SBCM'07*. Sao Paulo, Brasil, 2007.

[Pachet & Roy 07] F. Pachet, P. Roy, Exploring billions of audio features. In *Proc. of CBMI 07* Bordeaux, France, 2007.

[Pachet & Zils 04] F. Pachet, A. Zils, Automatic Extraction of Music Descriptors from Acoustic Signals. In *Proc. of ISMIR'04*. Barcelona, Spain, 2004.

[Kapur et al. 05] A. Kapur, R. I. McWalter, G. Tzanetakis, New Music Interfaces for Rhythm-Based Retrieval. In *Proc. of ISMIR'05*. London, England, 2005.

[Dixon et al. 04] S. Dixon, F. Gouyon, G. Widmer, Towards the Characterisation of Music via Rhythmic Patterns. In *Proc. of ISMIR'04*. Barcelona, Spain, 2004.

[Tsunoo et al. 09] E. Tsunoo, G. Tzanetakis, N. Ono, S. Sagayama, Audio Genre Classification by Clustering Percussive Patterns. In *Proc. of ASJ Spring Meeting*. 2009.

[Nakano et al. 04] T. Nakano, J. Ogata, M. Goto, Y. Hiraga, A drum pattern retrieval method by voice percussion. In *Proc. of ISMIR'04*. Barcelona, Spain, 2004.

[Pachet 00/2] F. Pachet, Rhythm as emerging structures. In *Proc. of ICMC'00*. Berlin, Germany, 2000.

[Toussaint 02] G. Toussaint, A Mathematical Analysis of African, Brazilian and Cuban *Clave* Rhythms. In *Proc. of BRIDGES: Mathematical Connections in Art, Music and Science*. Towson, 2002.

[Toussaint 04] G. Toussaint, The Geometry of Musical Rhythm. In *Proc. Japan Conference on Discrete and Computational Geometry*. Japan, 2004.

[Voyager 93] G. E. Lewis, *Voyager: Improvised Duos Between Human and Computer Improvisers*, Avant, (1993).

[Cypher 94] R Rowe, Shells for tarogato and interactive music system (1994). In *CD-ROM: Machine Musicianship*, Track 10, The MIT Press, Cabridge, 2001.