

USING PATTERNS TO GENERATE RHYTHMIC ACCOMPANIMENT FOR GUITAR

*Marcio Dahia, Hugo Santana
Ernesto Trajano, Geber
Ramalho*

Centro de Informática (UFPE)
Caixa Postal 7851
CEP 50732-970
Recife, PE, Brazil
mlmd,hps,etl,glr@cin.ufpe.br

Carlos Sandroni

Depto. de Música (UFPE)
Av. Prof. Moraes Rego, 1235
CDU, Recife, PE, Brazil
CEP: 50670-901
sandroni@terra.com.br

Giordano Cabral

LIP6
8 Rue du Capitaine Scott
75015
Paris, France
Giordano.CABRAL@lip6.fr

ABSTRACT

This work presents the model of a system that generates guitar rhythmic accompaniment of a song, given its chord grid and melody. In order to minimize the effects of the lack of formal knowledge which inherently exists in this musical dimension, the system uses a very natural approach in music: the contextualized reuse of rhythmic patterns, found in performances of famous musicians. To accomplish this task, two techniques of artificial intelligence were combined: case-based reasoning, for the modeling the "musical memory" (association of rhythmic patterns and the context where it should be used) and rule-based reasoning, to associate abstract intentions to the contextual characteristics of the patterns. As a case study, we developed *Cyber-João*, a program that generates a rhythmic accompaniment for Bossa Nova by chaining and adapting rhythmic patterns found in classic records of João Gilberto. Finally, the model was empirically compared with some other approaches implemented to solve the problem, demonstrating very motivating results.

1. INTRODUCTION

Although its potential use in programs for musical accompaniment and composition, the automatic generation of rhythm has not been much explored in the Computer Music literature. This task is difficult to model due to the lack of formal knowledge on this musical dimension: the musicians explain its rhythmic choices in high level, from abstract criteria such as swing, not being able to supply objective rules that elucidate their decisions in granularity of notes.

Moreover, in contrast to tasks such as harmony generation, which are facilitated by the availability of an extensive bibliography in musical theory, there are just few works on rhythm as accompaniment [1] [2]. In fact, the available literature itself indicates that this musical dimension is more associated with subjectivity.

These problems become obvious when we try to mimic the behaviour of instruments in which rhythm has an important role. This is the case of D'Accord Guitar¹ [3], an application developed to assist users in the task of learning how to play guitar, which shows the music been played in real-time in a virtual guitar. This system provides two methods for the rhythmic

accompaniment generation: selecting a rhythmic pattern from a list and using it during the whole song, or creating it from scratch by means of a computer keyboard. Neither of these approaches completely satisfies the user needs. Whereas the former is very poor rhythmically and totally different from a human-created accompaniment; the second is extremely time-consuming.

In this circumstance, automatic generation of rhythmic accompaniment could be used to guarantee good musical results without demanding any effort from the user. This task is one of those implemented by Automatic Accompaniment Systems. These systems generate musical lines (melody, chords, rhythm, etc.) of a given song using its chord grid, the other musicians' performance and the very system's previous performance. Among these systems, we highlight *ImPact* [7], which represents an earlier and successful effort of our research team in creating jazz bass lines in real time, by reusing melody fragments previously played by human musicians.

In this work, we discuss the reuse of the model implemented in *ImPact* to deal with the rhythmic accompaniment problem, as well as the necessary adaptations to cope with the domain specific features, mainly the system's inputs, musical fragments and output. In order to validate the work, we developed *Cyber-João*, an agent that generates Bossa Nova rhythmic accompaniment based on rhythmic patterns played by João Gilberto.

The paper is structured as follows. In Section 2, we discuss the complexity of the rhythmic accompaniment generation task. In Section 3, it is given a brief overview of the existing solutions, emphasizing the fragment reuse methodology. In Section 4, we present our model, explaining the most important development steps. In Section 5, we explain the experiment we made and the obtained results. Finally, in Section 6, we point out some conclusions and future works.

2. MODELING RHYTHMIC ACCOMPANIMENT

Generating rhythmic accompaniment is a difficult problem for several reasons. First, the accompaniment agent's environment is complex, since it is dynamic, non-deterministic, continuous and non-episodic [8]. If

¹ <http://www.daccordmusic.com>

the accompaniment must be performed in real time, the task is even harder.

Second, rhythmic accompaniment is under constraint, since although there are musical restrictions to guide the musicians; they are not enough to establish the rhythm to be played. In fact, a musician can play completely different rhythms, all of them acceptable.

Third, knowledge formalization is another problem, since musicians are often unable to explain precisely why they decide to play certain notes instead of other ones. This is particularly difficult in rhythm generation. In fact, contrary to the large literature on harmony and counterpoint and their use by computer systems, there are few works describing rhythm choices, most of them applied to drums [1] [2]. Rhythm generation seems to be rather intuitive than formally justifiable.

3. STATE OF ART

There are two major approaches to handle the automatic accompaniment generation problem: attempting to develop a computational model to generate notes from scratch, e.g., using grammars [5], or creating new accompaniments by appending music fragments, retrieved from a library, to one another. We have adopted the second approach whose advantages we describe next.

3.1. Fragment reuse

The motivation of reusing fragments is fourfold [7]. First, it minimizes the formalization problem, described in section 2, due to the fact that the fragments by themselves embody a certain musical knowledge. Second, fragment reuse is a scalable approach, since it is possible to use the same process to any instrument and style. Third, this approach improves expressiveness, since, by reusing fragments directly captured from a human musician, performance nuances can be taken into account. Fourth, this is a very natural approach to the rhythmic accompaniment problem; since it is very common practice to learn a rhythm by assimilating its most important rhythmic patterns [9].

However, in order to properly reuse fragments, there are five main topics that should be considered:

Should one use fixed-length or variable-length fragments? Using fragments of fixed length, makes the appending of fragments easier, however, it is not musically plausible;

What is the fragments granularity? The more fine-grained the fragment is, the more responsive the system can be, although, it can diminish the musical continuity;

What kind of description is necessary to index the fragments? The more attributes exist (e.g., tempo, density, dissonance, etc.), the more precise will be the choice of the fragments, although it will increase the dependency of style-specific knowledge and restrict the insertion of new fragments in the library;

Which are the best retrieval criteria? A rich description requires a more powerful technique (such as

similarity measures) than a simple one. For example, a random choice does not require any kind of description;

How to modify a fragment towards a better fit in the new context? Complex adaptations are hard to implement since the various characteristics of a given fragment are interdependent.

In the following, we discuss how two successful systems addressed these questions.

3.2. ImPact

The ImPact system [7] simulates a bass player. It deals with the problems described in section 2 by conceiving an innovative knowledge-intensive agent whose environment is the chord grid, its previous execution and the other musicians' performance.

It reuses 256 bass lines fragments with variable lengths, corresponding to typical chord sequences, such as II-V and II-V-I. Fragments are retrieved using a Case-Based Reasoning approach [6] supported by a rich description of the environment and the fragments' musical properties. Production Rules [8] are used to determine the musical properties the retrieved fragments are supposed to exhibit. The retrieval strategy, coupled with the rich description described above, generates excellent musical results. However, this approach is highly domain-dependent and difficult to implement. Regarding the adaptation of the retrieved fragments, some basic mechanisms (transposition, note deletion and insertion, etc.) are provided.

3.3. Band-in-a-box

Band-in-a-Box² is a broadly used commercial tool for automatically creating accompaniment and solo to electric guitar, piano, bass, drums and strings in a large number of styles (although the user may create his/her own ones). It has a number of musical fragments (up to 1620 per style) of different length (1 to 4 measures) using single chord granularity. In Band-in-a-box, there is no attributes concerning musical property of the fragments, just environment ones. Concerning the retrieval criteria, the software uses a random choice biased by user-entry weights to each fragment; using a simple set of rules to restrict the group of fragments that disagree with the environment attributes. For instance, there are some fragments that can only be played in the beginning of the song; others are supposed to be played in the end, and so on. Just transposition adaptation is provided.

4. OUR MODEL

Although the Band-in-a-Box approach is very attractive because of its simplicity and flexibility, we were most interested in musical plausibility, and the ImPact system was more qualified regarding this aspect.

² Band-in-a-Box is a trademark of PG. Music Inc. (<http://www.bandinabox.com>) all information obtained from user manual and using the software.

In order to evaluate in what extent ImPact approach could be adapted to the automatic generation of guitar rhythmic accompaniment, we chose a specific musical style: Bossa Nova. The advantages of choosing such a well-known and documented style is to benefit from the knowledge concerning rhythmic patterns description [4], as well as to better evaluate the system performance. In this context, we decided to reuse rhythmic fragments played by João Gilberto, since they are the most representative ones in Bossa Nova [9].

Due to the change of instrument (from bass to guitar), style (from Jazz to Bossa Nova) and environment (from ensemble to voice-and-guitar), the main implemented modifications of the original ImPact were: new library of musical fragments, new attributes of these fragments, new rules to improve the fragments retrieval and new perceptions (acquired from melody). It is important to emphasize that some of these new elements are difficult to elicit (as compared to the case of Jazz bass lines) due to the poor formalization of rhythmic accompaniment for guitar.

4.1. Rhythmic fragments

The task of choosing Bossa Nova rhythmic fragments was simplified by two musicology works [4] [9]. They transcribed and analyzed João Gilberto's most important rhythmic patterns and highlighted some important features:

All patterns last two measures (in 2/4 signature);

Just two kinds of events are allowed (one produced by the thumb and the other, produced by forefinger, middle finger, and ring finger together, which we call, from now on, "plucking block");

Several patterns anticipate their first event producing syncopation, an important rhythmic characteristic of Bossa Nova style.

There are four groups of patterns: the cyclic, beginning, special and fill-in, as depicted in Figure 1 (thumb in the lower clef and plucking block attacks in upper one);

There is a clear difference in performance of slow Bossa Nova songs (less than 120 quarter notes per minute) and faster ones, some patterns being forbidden to be used in the former.

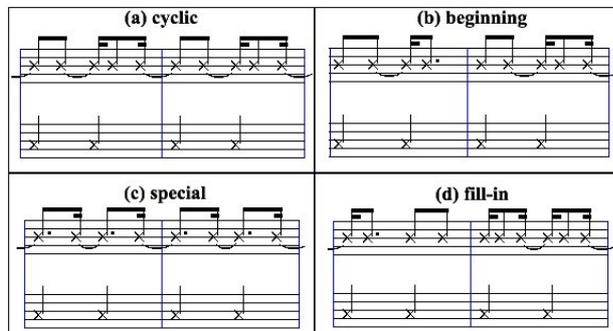


Figure 1. Examples of rhythmic pattern groups found in Bossa Nova.

Based on the above considerations, we decided that our rhythmic patterns would have fixed length and two-measure granularity. For the same reason, we did not implement adaptations on the Bossa Nova patterns: they will be reused exactly as played by João Gilberto.

Each pattern is represented by a small but effective set of attributes. They are of two kinds: environmental and musical ones. Environmental attributes describe the context where the pattern has been used. They are:

Harmonic rhythm, indicating how harmony (chords) changes in a given period of time (e.g. one or two measures). This attribute can have 15 different values ranging from a single chord lasting two 2/4 measures to four chords (one per beat);

Tempo, which can assume 2 values: "slow" (< 120 quarters per minute) or "fast".

Musical attributes describe the pattern's musical properties, i.e., its characteristics. They are:

Beginning, a binary attribute pointing out whether a pattern started at the down beat;

Fill-in, binary attribute indicating whether the pattern has been used as a "fill-in accompaniment", i.e., fragments commonly used when there is no melody been sung, or in turnarounds and turn backs.

Usage, determining how frequently the pattern has been used. The values range from 5 (maximum usage) and 1 (minimum usage);

Density, describing the number of musical events in each pattern's measure. The possible values are "High", "Medium", and "Low".

Table 1 shows an example of rhythmic pattern and its description. We have identified and indexed, with the help of experts and the literature [4] [9], 21 patterns in the library.

4.2. Retrieval technique

Following the ImPact approach, the retrieval technique consists in a mixed use of Rule-Based and Case-Based Reasoning. The former is used to extract from the environment (in our case, the chord grid and the melody) characteristics that suggest a musical intention (patterns' musical properties), as exemplified in Figure 2. To achieve these rules, we used the traditional method of knowledge acquisition [8] in which we interviewed specialists obtaining, after analysis, 6 rules that were consensus for all of them. Case-Based Reasoning is employed to effectively choose the best pattern in the library (case base).

In order to improve the retrieval, all attributes are weighted from 5 to 1, according to their importance, again obtained through specialists, as normally used in Case-Based systems. The query is performed by k-nearest neighbors [6].

Rhythmic Pattern					
Environment			Musical property		
Attribute	Value	Weight	Attribute	Value	Weight
Harmonic rhythm	"chord change in first and third beats"	4	Density	Medium	1
			Usage	5	2
Tempo	Fast or slow	5	Fill-in	No	3
			Beginning	No	5

Table 1. Example of a pattern and its attribute values.

<p>Rule: Fill-in in first measure</p> <p>IF (melody in the last beat of the second measure \leq 2 notes)</p> <p>AND (There is a V7-I chunk in the last beat of the second measure and the next one)</p> <p>THEN "Use same musical properties of the last pattern retrieved" Fill-in = "Yes"</p>
<p>Rule: Handle the music beginning</p> <p>IF (The music is beginning)</p> <p>THEN Beginning = "Yes" Density = "Low"</p>

Figure 2. Examples of two Cyber-João's Rules

5. EXPERIMENTS AND RESULTS

In order to analyze the real effectiveness of our approach, we have implemented Cyber-João. Also, to improve this analysis, we have implemented two other systems (less complex than the former): Crazy-João and João-in-a-Box. The first one selects the patterns to be used in the song by random choice; it is the baseline of our experiment. The second one, use the same retrieve criteria found in Band-in-a-Box, which is random choice bounded by weights and constrained by rules. The great benefit about these two new approaches is that they are not related with the style and demand little programming effort, instead of the approach used in ImPact/Cyber-João. So, we evaluated here if a knowledge intensive approach is really better in this case.

Thanks to modularity, it was possible to reuse a great amount of Cyber-João's code in implementation of Crazy-João and João-in-a-Box. However, it is important to note that there is a subtle difference on the effect of the set of rules between Cyber-João and João-in-a-Box. Whereas the former acts on the musical properties of the pattern being retrieved, the second acts directly in the rule base, restricting the groups to be used in a given moment (like Band-in-a-box).

We used four classical Bossa Nova songs: *Chega de Saudade*, *Desafinado*, *Insesatez* and *Lígia*. The election of the songs obeyed two criteria: Availability of a good-quality melody in MIDI format and the tempo of the songs (the two first ones are considered fast, while the other two, slow ones). Each song was generated by each three systems, resulting in a corpus of twelve samples. The corpus was blinded and presented to six evaluators, all of them, Bossa Nova musicians. They answered to a questionnaire in which they had to point out a score (among bad, acceptable, good, and excellent) for each sample and then, the best version for each song (ties were allowed).

Table 2, shows the score summarization grouped by system. Based on this data, we can perceive a clear difference between the Crazy-João results and the other's. While less than 9% of the evaluations were at least good for this system, the others received at least more than 66%. Obviously, as one could imagine, the use of random choice to select the rhythm patterns in the Bossa Nova style is completely discouraged.

System	Bad	Acceptable	Good	Excellent
<i>Crazy-João</i>	12	10	2	0
<i>Cyber-João</i>	2	4	7	11

Table 2. Example of a pattern and its attribute values.

The Table 3 summarizes the results of the best version for each song. There were three ties: two in *Desafinado* and one in *Insensatez*, so the sum of the total goes to 27 rather than 24. The results show that Cyber-João was evaluated as the best version more than 62% of the times; almost two times Joao-in-a-box did.

Song	<i>Crazy-João</i>	<i>João-in-a-box</i>	<i>Cyber-João</i>
<i>Desafinado</i>	0	3	5
<i>Chega de Saudade</i>	0	2	4
<i>Insensatez</i>	1	1	5
<i>Lígia</i>	0	3	3
TOTAL	1	9	17

Table 3. Results of the best version for each song.

6. CONCLUSIONS

In this paper, we presented a model for generating the rhythmic accompaniment for guitar, based on the reuse of previously stored rhythmic fragments. To validate this model we implemented Cyber-João, a system that plays Bossa Nova. It was coded in C++ and is fully integrated to D'Accord Guitar. It takes as input a dv3 file (D'Accord Guitar format) containing the melody and chord grid of a given song. Reusing a rhythmic pattern library, Cyber-João generates the correspondent accompaniment. It is important to note however, that this is a general approach to the rhythmic generation

problem. According to musicians' analysis of the preliminary outcome, Cyber-João exhibits excellent results: it was evaluated as good and excellent in 75% of the cases.

We are now coding a large corpus of Bossa Nova songs in order to provide a more precise evaluation of Cyber-João. In this perspective we intend to develop a feedback support system to help experts to judge and comment agent's choice and automatically generating statistical results, which can be used to improve the agent's knowledge.

The preliminary results encourage us to reuse the same approach in other musical styles. This is reinforced by the fact that the rhythmic fragment attributes in Cyber-João library are quite generic. Besides, the combination of this approach with an automatic extractor of rhythmic pattern [10] would speed up the development of new guitar accompaniment systems.

7. REFERENCES

- [1] Baggi, D. (1992). *NeurSwing: An Intelligent Workbench for the Investigation of Swing in Jazz*. Computer-Generated Music. D. Baggi, IEEE Computer Society Press: 79-93.
- [2] Burton, A. R., Vladimirova, T. (1997). Genetic Algorithm Utilizing Neural Network Fitness Evaluation for Musical Composition, ICANNGA.
- [3] Cabral, G. et al (2001). D'Accord Guitar: an innovative guitar performance system In proc. *Journées d'Informatique Musicale*. Bourge, France.
- [4] Garcia, Walter. (1999). *Bim Bom: A contradição sem conflitos de João Gilberto*. Ed. Guerra e Paz.
- [5] Johnson-Laird, P. N. (1991). Jazz Improvisation: A Theory at the Computational Level In: P. Howell, R. West, and I. Cross, eds, *Representing Musical Structure*. Academic Press.
- [6] Kolodner, J. (1993). *Case-Based Reasoning*. Ed. Morgan Kaufmann.
- [7] Ramalho, G., Rolland, P.-Y., & Ganascia, J.-G. (1999). An Artificially Intelligent Jazz Performer. *Journal of New Music Research*. 28(2). pp. 105-129. Swets & Zeitlinger: Amsterdam.
- [8] Russell, S & Norvig, P (1995). *Artificial Intelligence: a modern approach*. Ed. Prentice-Hall.
- [9] Sandroni, C. (1988). *O Olhar do Aprendiz: Observações sobre a prática do violão popular no Brasil, fartamente documentada por exemplos em partitura*. (Unpublished paper).
- [10] Santana, H. et al (2003). VexPat - An Analysis Tool for the Discovery of Musical Patterns. In proc. SBCM'2003.