Generating Rhythmic Accompaniment for Guitar: the Cyber-João Case Study

Márcio Dahia, Hugo Santana, Ernesto Trajano, Carlos Sandroni* and Geber Ramalho

Centro de Informática and Departamento de Música*
Universidade Federal de Pernambuco (UFPE)
Caixa Postal 7851 - CEP 50732-970 – Recife – PE – Brasil

{mlmd, hps,etl,glr}@cin.ufpe.br, sandroni@terra.com.br

Abstract. *Despite its potential use in automatic accompaniment software, guitar rhythm generation has not been discussed in computer music literature to our knowledge. This task is difficult to model due to the sketchiness of knowledge concerning this musical dimension. This paper presents an agent that generates guitar rhythmic accompaniment of a song, given its chord grid and melody. The adopted approach is the contextualized reuse of rhythmic fragments extracted from human musicians' previous performance. As a case study, we developed Cyber-João, an agent that plays Bossa Nova reusing excerpts of João Gilberto's classic records. The achieved results encourage the use of this approach in other musical styles.*

1. Introduction

D'Accord Guitar [Cabral 2001; Cabral et al. 2001] is an application developed to assist users in the task of learning how to play guitar. It has two main modules. First, the player, which shows the execution of the song in a virtual guitar fingerboard displayed on the computer screen, while indicating the lyrics position and the chord to be played at the same moment. Second, the editor, used to create new songs in D'Accord file format (dv3 format).

In the first version of this system, two methods of generating a rhythm are provided: 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 context, 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 the *automatic accompaniment systems*. In fact, 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 [Ramalho et al. 1999], 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 ImPact model 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 achieved results, reusing a base composed by 21 rhythm patterns, were quite impressive according specialists.

The paper is structured as follows. In Section 2, we discuss the complexity of the rhythmic accompaniment generation. 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. And in Section 5, we point out some conclusions and future works.

2. Modeling Rhythmic Accompaniment

Generating rhythmic accompaniment is hard problem for various reasons. First, the accompaniment agent's environment is complex, since it is dynamic, non-deterministic, continuous and non-episodic [Russell 1995]. If the accompaniment must be performed in real time, the task is even harder.

Second, as a creative design task, rhythmic accompaniment is under constraint, since, although there are musical restrictions, 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 answer 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 [Baggi 1992; Burton 1997]. Rhythm generation seems to be rather intuitive than formally justifiable.

3. State of Art

There are two major approaches to handle the automatic accompaniment/improvisation generation problem: attempting to develop a computational model to generate notes from scratch (e.g., using grammar-based [Johnson-Laird 1991]), or creating new accompaniments by appending music fragments, retrieved from a library, to one another [Band-in-a-Box 1998]. We have adopted the second approach whose advantages we describe next.

3.1 Fragment reuse

The motivation of reusing fragments is threefold [Ramalho et al. 1999]. First, it minimizes the formalization problem, described in section 2.1 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.

However, some important decisions shall be made in five main topics:

- 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;
- What are the best retrieval criteria? A rich description requires a more powerful technique (such as similarity measures) than a simple description. 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 [Ramalho 1997] 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 [Kolodner 1993] supported by a rich description of the environment and the fragments' musical properties. Production Rules [Russell 1995] 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

The Band-in-a-Box [Band-in-a-box 1998] 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 hundreds of musical fragments (up to 600 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. 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.

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 [Garcia 1999], 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 [Sandroni 1985].

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 (melody and chord grid). It is important to stress 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 extremely simplified by two musicology works [Garcia 1999, Sandroni 1988]. 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");
- Most patterns anticipate their first event producing syncopation, an important rhythm characteristic of Bossa Nova style;
- 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;

The general remark from these points is that the degrees of freedom are quite constrained in guitar rhythmic accompaniment (compared to Jazz bass lines).

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

Cyber-João possesses a small but effective set of pattern attributes, composed by 2 environment ones (determining the context where the fragment has been used) and 5 musical properties ones (determining the characteristics of the fragment). The environment attributes 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”;

The musical properties are:

- *Anticipated*, a binary attribute pointing out whether a fragment started at the down beat;
- *Next fragment anticipation*, which indicates (with “yes” or “no” values) whether the next fragment was anticipated;
- *Fill-in*, indicating whether and when the fragment 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. This attribute can assume 4 values (“first measure”, “second measure”, “both measures” or “none”);
- *Recurrence*, determining how frequently the fragment has been used (“high”, “medium”, “low”);
- *Density*, describing the number of musical events in each fragment’s measure. Some possible values are “high in the first measure, low in the second”, “very high in both measures”, etc.

Table 1 shows an example of rhythmic fragment (illustrated by thumb attacks, in the bass clef, and plucking block attacks, in treble clef) and its description. We have identified and indexed, with the help of experts and the literature [Garcia 1999], 21 fragments in the library.

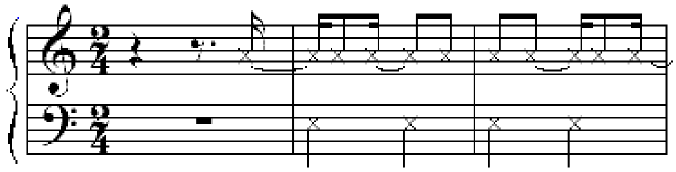
4.2 Retrieval technique

Following the ImPact approach, the retrieval technique consists in a mixed use of Production Rules 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 1. Up to now, there are 12 rules similar to this one. The second 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 0, according to their importance, as normally used in Case-Based systems. The query is performed by k-nearest neighbors [Kolodner 1993].

Table 1. Example of pattern and its attribute values

Rhythmic Pattern



Environment			Musical property		
Attribute	**Value**	**Weight**	**Attribute**	**Value**	**Weight**
Harmonic rhythm	At least, one chord change in the first measure and one in the second one	4	Anticipated	Yes	5
			Next fragment anticipation	Yes	4
Tempo	Fast songs	5	Fill-in	None	3
			Recurrence	High	2
			Density	Medium in first and second measure.	4

```
Rule: Fill-in in first measure
IF
     (There is no melody in the first measure) AND
     (The first chord of the measure is, a 7th one) AND
     (There is no other chord in that measure) AND
     (The first chord of the next measure resolves the tension)
THEN
     Density = "High in the first measure and medium in the second"
     Fill-in = "First measure"
```

```
Rule: Handle the music beginning
IF
     (The music is beginning)
THEN
     Anticipated = "No"
     Recurrence = "High"
     Fill-in = "No"
```

Figure 1. Example of two rules

5. Results and Conclusions

We have implemented Cyber-João in C++ and fully integrated it 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.

According to musicologists' analysis of the preliminary results, Cyber-João exhibits excellent results. None of the 211 rhythmic patterns chosen to perform two classical Bossa Nova songs, *Desafinado* and *Insensatez* were considered inadequate or incorrect. These are only preliminary results, but they are impressive.

We presented a novel application of pattern reuse approach to generating rhythmic accompaniment for guitar. A preliminary validation has been performed by the implementation of an agent capable of creating Bossa Nova accompaniment.

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, automatically generating the 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 [Santana 2003] would speed up the development of new guitar accompaniment systems.

References

Baggi, D. (1992). NeurSwing: An Intelligent Workbench for the Investigation of Swing in Jazz. In D. Baggi (Eds.), *Computer-Generated Music* (pp. 79-93). IEEE Computer Society Press.

Band-in-a-box (1998). Band-in-a-box Pro 8.0. Canada: PG Music Inc

Burton, A. R., Vladimirova, T. (1997). Genetic Algorithm Utilising Neural Network Fitness Evaluation for Musical Composition, ICANNGA Abstracts, p. 23

Cabral, G. et al (2001). D´Accord Guitar: an innovative guitar performance system In *Journées d´Informatique Musicale*. Bourge, France.

Cabral, G. Zanforlin, I., Lima, R., Santana, H. & Ramalho, G. (2001). *Da Cifra Para o Braço: Estudo dos Problemas de Execução Musical em Violão e Guitarra.* In proc. VIII Brazilian Symposium on Computer Music - SBCM'2001. (pp. 934 - 940).

Garcia, Walter. (1999). *Bim Bom: A contradição sem conflitos de João Gilberto*. Ed. *Guerra e Paz.*

Santana, H., Dahia, M., Trajano, E. & Ramalho, G. (2003). VexPat - An Analysis Tool for the Discovery of Musical Patterns. In proc. IX Brazilian Symposium on Computer Music - SBCM'2003.

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.

Kolodner, J. (1993). Case-Based Reasoning. Ed. Morgan Kaufmann.

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.

Russell, S & Norvig, P (1995). Artificial Intelligence: a modern approach. Ed. Prentice-Hall.

Sandroni, C. (1988). *O Olhar do Aprendiz: Observações sobre a prática do violão popular no Brasil, fartamente documentada por exemplos em partitura*. (Unpublished paper)