



**Pós-Graduação em Ciência da Computação**

**“Gerando Acompanhamento Rítmico Automático  
para Violão: Estudo de caso do *Cyber-João*”**

**Por**

***Márcio Leal de Melo Dahia***

**Dissertação de Mestrado**



Universidade Federal de Pernambuco  
posgraduacao@cin.ufpe.br  
[www.cin.ufpe.br/~posgraduacao](http://www.cin.ufpe.br/~posgraduacao)

RECIFE, MARÇO/2004



UNIVERSIDADE FEDERAL DE PERNAMBUCO  
CENTRO DE INFORMÁTICA  
PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

MÁRCIO LEAL DE MELO DAHIA

“Gerando Acompanhamento Rítmico Automático para  
Violão: Estudo de caso do Cyber-João”

*ESTE TRABALHO FOI APRESENTADO À PÓS-GRADUAÇÃO EM  
CIÊNCIA DA COMPUTAÇÃO DO CENTRO DE INFORMÁTICA DA  
UNIVERSIDADE FEDERAL DE PERNAMBUCO COMO REQUISITO  
PARCIAL PARA OBTENÇÃO DO GRAU DE MESTRE EM CIÊNCIA DA  
COMPUTAÇÃO.*

ORIENTADOR: Geber Lisboa Ramalho  
CO-ORIENTADOR: Carlos Sandroni

RECIFE, MARÇO/2004

**Dahia, Márcio Leal de Melo**

**Gerando acompanhamento rítmico automático  
para violão : estudo de caso do Cyber-João / Márcio  
Leal de Melo Dahia. – Recife : O Autor, 2004**

**94 folhas : il., fig., tab., quadros.**

**Dissertação (mestrado) – Universidade Federal  
de Pernambuco. Cln. Ciência da Computação, 2004.**

**Inclui bibliografia e anexos.**

**1. Ciência da computação – Inteligência artificial.  
2. Computação musical – Sistemas de  
acompanhamento automático . 3. Violão – Bossa  
Nova – Obras de João Gilberto – Padrão rítmico. 4.  
Acompanhamento rítmico – Raciocínio baseado em  
caso – Regras de produção. I. Título.**

**004.855.5  
006.31**

**CDU (2.ed.)  
CDD (22.ed.)**

**UFPE  
BC2005-270**

*A meus pais, irmãos e sobrinhos,*

*a Marcela,*

*a meu avô (em memória).*

## **Agradecimentos**

Em primeiro lugar, agradeço a meus pais Alberto Antonio Dahia e Mariza Leal de Melo Dahia, por sempre estarem dispostos a prover absolutamente tudo que é necessário para um ser humano. Amor, apoio e incentivo nas horas difíceis são apenas algumas delas.

Só conheço quatro pessoas que tem a sorte que eu quando se trata de pais: meus irmãos Patrícia, Beto, Sandra e Fábio, aos quais também agradeço pela ajuda incondicional, pelas conversas divertidas e também pelas sérias.

A Marcela, que sem dúvida, foi o principal motivo da minha permanência em Recife, no tempo em que tudo dava errado. Além de grande ajuda na redação e correção dessa dissertação.

Ao meu orientador e amigo Geber Ramalho por acreditar em minha capacidade dar continuidade ao mestrado quando poucos acreditavam, pelo companheirismo e competente trabalho de orientação.

Aos membros da banca examinadora Edílson Ferneda e Flávia Barros pelas contribuições valiosas dadas ao trabalho. Serei eternamente grato a Flávia pela grande ajuda nos tempos difíceis no início do mestrado. Ela, Geber e Augusto Sampaio foram os responsáveis pela minha permanência no curso.

Por fim, a toda (ex-)equipe de Computação Musical do CIn, os amigos Giordano, Bel, Mouse. Em especial, agradeço a Hugo Santana e Ernesto Trajano pela ajuda no desenvolvimento do projeto e correção dos capítulos dessa dissertação.

## RESUMO

Apesar de seu potencial uso em programas de acompanhamento e composição musicais, a geração automática de ritmo não tem sido muito discutida na literatura de computação musical. Essa tarefa é bastante difícil de modelar devido ao pouco conhecimento formal sobre essa dimensão musical: os músicos explicam suas escolhas rítmicas em alto nível, a partir de critérios abstratos como swing, por exemplo, não sendo capazes de fornecer regras objetivas que elucidem suas decisões em granularidade de notas. Além disso, ao contrário de tarefas como a geração de harmonia, que são facilitadas pela disponibilidade de uma extensa bibliografia em teoria musical, pouco existe sobre ritmos como acompanhamento. De fato, a própria literatura disponível deixa a entender que essa dimensão musical é mais associada à subjetividade.

Este trabalho apresenta um sistema que gera acompanhamento rítmico para violão automaticamente. Para minimizar os efeitos da dificuldade de formalização do conhecimento, o sistema faz uso de uma abordagem bastante natural em música, o reuso contextualizado de padrões rítmicos encontrados em performances de músicos de renome. Para isso, foram combinadas duas técnicas de inteligência artificial: raciocínio baseado em casos, para a modelagem da “memória musical” (associação de padrões rítmicos e o contexto onde devem ser usado) e raciocínio dedutivo baseado em regras, para associar intenções abstratas às características contextuais dos padrões.

Como estudo de caso, foi desenvolvido o Cyber-João, um programa capaz de gerar acompanhamento rítmico para Bossa Nova encadeando e adaptando padrões rítmicos encontrados em gravações clássicas de João Gilberto. Por fim, o modelo do Cyber-João foi comparado empiricamente com algumas outras abordagens implementadas para a resolução do problema, demonstrando um desempenho bastante satisfatório.

**Palavras-chaves:** geração automática de ritmo, violão, Bossa Nova, raciocínio baseado em casos, raciocínio baseado em regras.

## ABSTRACT

Although its potential use in programs of musical accompaniment and composition, the automatic generation of rhythm has not been argued 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, for example, not being able to supply objective rules that elucidate his/her decisions in granularidade 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 rhythms as accompaniment. In fact, the available literature itself indicates that this musical dimension more is associated with the subjectivity.

This work presents a system that generates automatically rhythmic accompaniment for guitar. To minimize the effect the lack of formal knowledge, 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 had been combined: case based reasoning, for the modeling the "musical memory" (association of rhythmic patterns and the context where it must be used) and rules based reasoning, to associate abstract intentions to the contextual characteristics of the patterns.

As 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 *Cyber-João's* model was empirically compared with some other approaches implemented to solve the problem, demonstrating very motivating results.

**Keywords:** automatic generation of rhythm, guitar, Bossa Nova, case based reasoning, rules based reasoning.

## Sumário

1 . Introdução	1
1.1. Objetivos	2
1.2. Organização da Dissertação	3
2 . Motivação e Dificuldades	4
2.1. O Ritmo	4
2.2. O Acompanhamento de Violão Popular	5
2.3. D´Accord Violão	6
2.3.1. D´Accord Violão <i>Player</i>	6
2.3.2. D´Accord Violão Editor	8
2.4. Limitações da Edição de Ritmos no D´Accord Violão	9
2.5. Sistemas de Acompanhamento Automático	11
2.5.1. O Problema da Formalização do Conhecimento Musical nos SAA	12
2.5.2. Dificuldades na Abordagem de Agentes	14
2.6. Decisões a serem tomadas pelos SAA	15
2.7. Peculiaridades do Acompanhamento Rítmico para o Violão	17
3 . Estado da Arte	20
3.1. Contexto do Problema	20
3.2. Abordagem nota por nota	23
3.2.1. Sistemas baseados em Gramáticas	23



3.2.2. Algoritmos Genéticos	25
3.3. Reuso de Fragmentos Musicais	26
3.3.1. <i>Band-in-a-Box</i>	30
3.3.2. ImPact	32
3.4. NeurSwing	43
3.5. Considerações Finais	44
4 . O Modelo Proposto	46
4.1. Escolhendo a Metodologia a ser Adotada	46
4.2. Escolhendo o Estilo Musical: Bossa Nova	48
4.3. Escolhendo os Padrões Rítmicos	49
4.4. Representação e Natureza dos Padrões	54
4.4.1. Atributos Associados aos Padrões	54
4.4.2. Técnica de Recuperação de Padrões Utilizada	58
4.5. Considerações Finais	60
5 . <i>Cyber-João</i> : Implementação e Resultados	61
5.1. Visão Geral	61
5.2. Seletor de ritmo: Motor de Raciocínio Baseado em Casos	64
5.3. Metodologia de Testes	66
5.4. Protocolo Experimental	69
5.4.1. Resultados da música <i>Desafinado</i>	70
5.4.2. Resultados da música <i>Chega de Saudade</i>	71

5.4.3. Resultados de música <i>Insensatez</i>	71
5.4.4. Resultado da música <i>Lígia</i>	72
5.4.5. Resultados Gerais	73
5.5. Considerações Finais	74
6 . Conclusão	75
6.1. Contribuições e Dificuldades Encontradas	76
6.2. Trabalhos Futuros e em Andamento	77
7 . Referências Bibliográficas	80
Anexo A – Lista de Padrões Rítmicos Disponíveis	85
Anexo B – Lista das Regras	92
Anexo C – Questionário de Avaliação dos Sistemas	94

## Lista de figuras

Figura 1 – Tela principal do D’Accord Violão <i>Player</i>	7
Figura 2 – Processo de gravação do D’Accord Violão Editor dividido em passos – Adaptado de (Cabral 2002)	8
Figura 3 – Ilustração da aplicação do ritmo aos acordes – Extraído de (Cabral 2002)	9
Figura 4 – Grade de acordes de “Stella by Starlight” (de N. Washington and V Young) – Extraído de (Ramalho 1997)	12
Figura 5 – Quatro improvisações sobre Dm7  A7 encontradas em Miles Davis – Extraído de (Ramalho 1997)	13
Figura 6 – Contextualização do problema de acompanhamento rítmico para violão em relação a Computação Musical	21
Figura 7 – Diagrama de transição de estados da gramática de geração da linha de contrabaixo	24
Figura 8 – Exemplo de árvore probabilística derivacional para ritmo (pesos em %) – Extraído de (Ramalho 1997)	25
Figura 9 – Edição de Estilo no <i>Band-in-a-Box</i>	32
Figura 10 – Ilustração do ImPact como agente inteligente	33
Figura 11 – Funcionamento do ciclo RBC, os quatro <i>REs</i>	36
Figura 12 – Exemplo de Memória Plana	38
Figura 13 – Exemplo de rede de atributos compartilhados	38
Figura 14 – Principal padrão rítmico criado por João Gilberto. A clave de sol indica os ataques das puxadas e a clave de fá indica os ataques do bordão.	50
Figura 15 – Erro comumente praticado por iniciantes: executar um evento de bordão no ultimo quarto de tempo.	51

Figura 16 – O padrão rítmico justaposto a ele mesmo	51
Figura 17 – Padrão cíclico resultante do padrão clássico	51
Figura 18 – Exemplo de padrão resultante da combinação dos padrões das Figura 14 e Figura 17	52
Figura 19 – Padrão rítmico especial. Utilizado em momentos de ritmo harmônico alto	53
Figura 20 – Padrão rítmico especial. Utilizado em momentos de ritmo harmônico alto	53
Figura 21 – Exemplo de padrão de virada	54
Figura 22 – Ilustração do emprego da regra virada (avaliação das condições).	59
Figura 23 – Ilustração do emprego da regra virada (escolha do padrão)	60
Figura 24 – Diagrama esquemático arquitetural do nosso modelo	61
Figura 25 – Diagrama de classes do motor RBC (UML)	65
Figura 26 – Diagrama de funcionamento do programa proposto	78
Figura 27 – Tela principal do <i>VexPat</i>	79

## Lista de tabelas

Tabela 1 – Resumo dos sistemas citados no texto	45
Tabela 2 – Eventos executados por quarto de tempo	50
Tabela 3 – Todos os valores possíveis de Ritmo Harmônico associados à presença de acordes nos tempos dos compassos.	56
Tabela 4 – Pesos associados ao atributos	57
Tabela 5 – Exemplo de atributos associados a um padrão rítmico.	58
Tabela 6 – Resultados da avaliação das amostras de Desafinado	71
Tabela 7 – Resultados da avaliação das amostras de Chega de Saudade	71
Tabela 8 – Resultados da avaliação das amostras de Insensatez para cada avaliador	72
Tabela 9 – Resultados da avaliação das amostras de Lígia para cada avaliador	72
Tabela 10 – Totalização do quesito melhor resultado	73

## Lista de quadros

Quadro 1 – Dois exemplo de regras implementadas	59
Quadro 2 – Todas as regras utilizadas no <i>João-in-a-box</i>	69

## 1. Introdução

Desde o surgimento da música, o homem tem buscado analisá-la com um certo rigor formal. É muito fácil encontrar fatos históricos que comprovem essa afirmação: o próprio surgimento do sistema musical tonal, principal base da música ocidental, atribuída a Pitágoras, o grande matemático grego, tem como principal fundamento a existência de uma relação de proporção entre as notas musicais (McClain 1994). Constata-se que a área de música, ao longo dos anos, tem se mostrado bastante à vontade em absorver novidades científicas e tecnológicas. Talvez por esse motivo, o surgimento da Computação Musical date da mesma época que a aparição dos primeiros computadores: a primeira composição algorítmica gerada por computador, trabalho de Hiller e Isaacson, é de 1959 (Roads 1995).

A partir daí, o interesse dos músicos pelo computador não parou de crescer. De fato, o computador permite, de forma inigualável, que o músico amplie seus horizontes *ad libitum* (Roads 1995).

Hoje, é possível encontrar programas de computador capazes de fazer as mais diversas tarefas musicais, como por exemplo, síntese de som, análise musical assistida por computador, composição produzida por computador e auxílio ao aprendizado de teoria musical e de instrumentos musicais. Vários esforços vêm sendo realizados no sentido de utilizar os recursos de multimídia dos computadores para aprimorar os processos de escrita, leitura e aprendizado musicais. Porém, ainda há carência de uma solução adequada.

Um esforço nesse sentido é o D'Accord Violão (Cabral 2002). Desenvolvido no Centro de Informática da Universidade Federal de Pernambuco, este software funciona como um tutor do tipo: “veja-ouça-e-repita”. Embora, represente um grande avanço em relação a uma grande quantidade de programas destinados ao ensino de instrumentos musicais, o D'Accord Violão ainda apresenta algumas limitações, principalmente na edição de músicas. Uma delas é a edição do

acompanhamento rítmico, que produz resultados monótonos e sem *swing*, quando gerado automaticamente.

A tarefa de produzir um acompanhamento rítmico semelhante ao tocado por um humano é bastante complexa, por dois motivos principais. O primeiro é pela falta de um conhecimento formal de como um executor produz o acompanhamento. Em muitos casos, o próprio executor é incapaz de explicar algumas de suas decisões (Johnson-Laird 1991). Este fato é vastamente analisado nos Sistema de Acompanhamento Automático (SAAs) (Ramalho 1994). Tais sistemas se caracterizam por gerar linhas musicais (melodia, ritmo ou acordes) a partir de uma grade de acordes, o que o sistema já tocou no passado e, no caso de uma produção coletiva, o que os outros músicos estão tocando no momento.

O segundo é que, embora o acompanhamento rítmico seja uma das peças fundamentais na performance de uma música no violão, guitarra ou outro instrumento harmônico qualquer, pouco tem sido discutido na literatura de Computação Musical especificamente sobre acompanhamento rítmico.

### 1.1.Objetivos

Nesse trabalho apresentamos um estudo sobre a geração automática de acompanhamento rítmico para o violão a fim de suavizar o problema do acompanhamento rítmico existente no D'Accord Violão.

Como estudo de caso, desenvolvemos o *Cyber-João*, um programa capaz de gerar uma linha de acompanhamento rítmico para o violão no estilo Bossa Nova, dadas a grade de acordes e a melodia de uma determinada música. Como abordagem de geração dessa linha, utilizamos uma combinação de raciocínio dedutivo baseado em regras com raciocínio baseado em casos. Tal abordagem já foi desenvolvida para o ImPact (Ramalho 1997), SAA criado para simular a performance de um baixista de Jazz tocando com sua banda. Os excelentes resultados obtidos e a vasta documentação disponível são alguns dos argumentos que justificam essa escolha. Porém, por causa das diferenças dos dois sistemas em relação a instrumento, estilo e finalidade, são necessárias diversas adaptações, principalmente nos fragmentos e na forma de indexá-los e recuperá-los.



O *Cyber-João* é alimentado com 21 padrões rítmicos extraídos de performances clássicas de João Gilberto (daí o nome do programa), além de 6 regras que influenciam a escolha dos padrões.

Para auxiliar no processo de avaliação do sistema, criamos alguns outros sistemas que servirão de comparativo. O primeiro gera a linha rítmica de forma totalmente aleatória servindo como referência de pior caso. Outro sistema também implementado utiliza um processo aleatório influenciado pelo peso associado a cada padrão.

Esperamos que, como consequência deste trabalho, possamos contribuir de alguma forma para a consolidação do conhecimento relacionado à geração automática de acompanhamentos rítmicos, cuja literatura ainda é bastante escassa.

## **1.2.Organização da Dissertação**

O restante da dissertação está organizado da seguinte forma: no capítulo 2 é apresentada a motivação do trabalho bem como as dificuldades relacionadas com o seu desenvolvimento. No capítulo 3 faremos um apanhado geral sobre sistemas de acompanhamento automático e tecnologias usadas para seu desenvolvimento. No capítulo 4 descrevemos as decisões de projeto adotadas, detalhes de implementação e os resultados preliminarmente obtidos. Finalmente, expomos as conclusões e sugestões para trabalhos futuros.

## 2. Motivação e Dificuldades

Neste capítulo, faremos uma introdução aos conceitos de ritmo e acompanhamento de violão. Apresentaremos o programa D'Accord Violão, seu processo de edição de músicas e discutiremos as limitações do mencionado processo no que se refere à edição de acompanhamentos rítmicos. Citaremos, também, as dificuldades em gerar acompanhamentos musicais automaticamente utilizando a abordagem dos Sistemas de Acompanhamento Automático de uma forma geral e problemas específicos do caso abordado nesse trabalho: a geração automática de acompanhamento rítmico para violão.

### 2.1.O Ritmo

O significado da palavra ritmo, do grego *rhythmos*, designa “aquilo que flui, aquilo que se move” (Keifer 1984). Segundo Keifer:

*O termo fluir deve ser tomado aqui em sentido mais geral possível. Platão disse: “Vós distinguireis o ritmo no vôo dos pássaros, nas pulsações das artérias, nos passos dos dançarinos, nos períodos de um discurso” (Keifer 1984).*

Uma observação importante é que a percepção do ritmo acontece justamente quando esse fluxo apresenta descontinuidade (Keifer 1984). Por exemplo, não percebemos ritmo na emissão contínua de um som, e sim, quando dividimos esse som em partes e o emitimos em sequência.

Há, no entanto, outra idéia ligada ao ritmo: a de ordem. De fato, as descontinuidades, citadas anteriormente, quando se sucedem caoticamente, provocarão sensação de confusão. Na realidade, quando se fala em ritmo, supõe-se sempre uma ordenação que implica uma certa regularidade (periodicidade) de elementos se não iguais, pelo menos comparáveis (Keifer 1984). Foi este fato que levou Aristóxeno, discípulo de Aristóteles, a definir o ritmo como “ordem na repartição das durações” (Keifer 1984).

No que se refere especificamente ao fluxo musical, os fatores que o geram são fundamentalmente dois: a intensidade e a duração dos sons.

O fator intensidade é explicado por Keifer da seguinte forma:

*Suponha agora, que um instrumento emita sons de igual duração, alternando sons fortes e fracos. Teremos aí um ritmo diferente de outro, por exemplo: forte, fraco, fraco [...] Se um instrumento tocasse sons da mesma intensidade, nós ouviríamos sons de intensidade desigual. É famosa a experiência do metrônomo: embora a intensidade seja a mesma em cada batida, ouvimos sempre o tic mais forte que o tac (Keifer 1984).*

O fator decisivo para a definição do ritmo musical é a idéia de duração. A percepção do ritmo musical surge da percepção recorrente de seqüências de notas mais rápidas ou mais lentas, formando o que se pode chamar de *padrão rítmico*.

## **2.2.O Acompanhamento de Violão Popular**

O violão é um instrumento extremamente popular no Brasil. Para alguns, chega a ser um “instrumento institucional”, tão importante para a nossa identidade cultural como o futebol e o carnaval (Sandroni 1985). Por esse motivo, apesar de existirem duas escolas distintas de violão, a clássica e a popular. A primeira, extremamente difundida pelo mundo, é a escola popular que é seguida pela imensa maioria dos violonistas brasileiros.

O violão firmou-se no decorrer do século XIX como principal instrumento acompanhador do canto popular brasileiro (Sandroni 1985). Entende-se por acompanhamento uma linha musical executada concomitantemente ao solista, que serve de suporte harmônico e rítmico a ele. Na Música Popular Brasileira, esse solista geralmente é a voz humana, criando um estilo bastante popular chamado *voz e violão*.

O acompanhamento no violão é tocado da seguinte forma (utilizando um como referência um instrumentista destro): com a mão esquerda o instrumentista escolhe que acorde deseja tocar (o componente harmônico do acompanhamento). Uma vez que um acorde pode ser tocado de várias formas diferentes, ele deve também escolher quais notas comporão efetivamente o acorde, isto é, em quais casas de quais cordas o instrumentista pressionará o dedo a fim de produzir a nota

desejada. Com a mão direita, o instrumentista decide quando o acorde deve soar, isto é, define o componente rítmico do acompanhamento.

### **2.3.D´Accord Violão**

D´Accord Violão é um software desenvolvido no Centro de Informática da UFPe para auxiliar no aprendizado de violão (Cabral 2001; Cabral 2002). Ele foi criado com o objetivo de preencher a grande lacuna existente na área de softwares para o aprendizado de peças musicais que sejam apropriados para notações musicais simples, como as músicas cifradas.

Seu princípio básico é apresentar a informação da maneira mais intuitiva possível, facilitando o aprendizado de acompanhamentos musicais no violão e na guitarra (Cabral 2002). O programa possui dois módulos principais, são eles: o D´Accord Violão *Player* e o D´Accord Violão Editor, que serão descritos a seguir.

#### **2.3.1.D´Accord Violão *Player***

É neste modulo onde os usuários efetivamente aprendem a tocar. Para atingir seu objetivo, o D´Accord Violão *Player* integra e estende, em sua interface gráfica, os princípios de dois tipos de programas que até então só existiam como softwares separados. São eles: os Sistemas de Simulação Instrumental (SSI), que exibem diretamente uma música sendo tocada em um instrumento virtual (Beck 1996), neste caso específico, o braço do violão; e os Sistemas de Karaokê, que sincronizam a exibição da letra da música com a sua melodia.

Além da sincronização da melodia em formato MIDI (Rothstein 1992) com a letra da música (que configura o karaokê), há também uma indicação de que acorde deve ser tocado no momento, onde se localizam as notas que o compõe no braço do violão e com que dedo cada nota deve ser tocada; como pode ser visto na Figura 1.

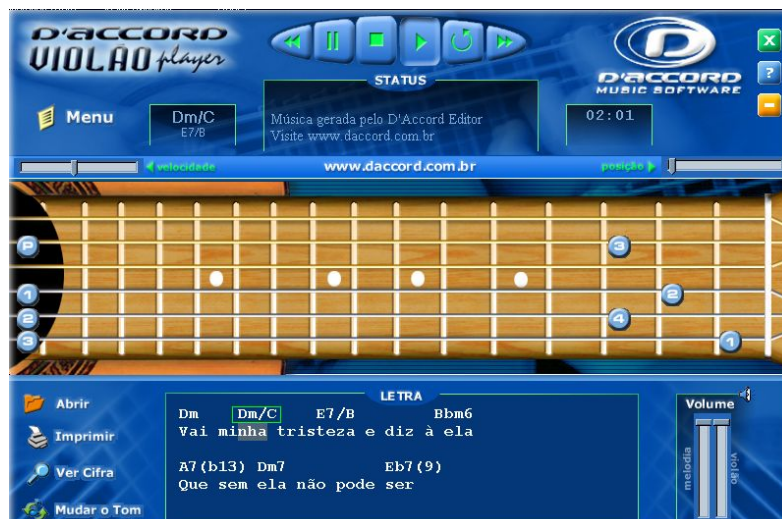


Figura 1 – Tela principal do D'Accord Violão *Player*

Destacamos ainda as seguintes funcionalidades:

- Visualização do braço do violão de quatro formas diferentes: de frente, na mesma posição do violão do usuário, visto de frente com cordas invertidas, e na mesma posição do violão do usuário com cordas invertidas, esses dois últimos, dedicados a usuários canhotos.
- Modificação do andamento da música para que este se adeque melhor ao estilo ou limitação do usuário.
- Repetição de um determinado trecho da canção indefinidamente.
- Aplicação de transposição aos acordes e à melodia da música.
- Exibição da melodia da música como notas no braço do violão e de solos existentes nela.

Dessa forma, o D'Accord Violão *Player* cria um ambiente semelhante ao de aprendizado por imitação de um professor, considerado como um dos melhores métodos pedagógicos para instrumentos musicais (Clarke 1993), mantendo, contudo, um certo grau de formalização estabelecido pela notação de cifras e pela marcação de andamento e outros conceitos, como compasso, tempo etc.

### 2.3.2.D´Accord Violão Editor

Nele, usuários com conhecimento musical mais avançado são capazes de criar (editar) suas músicas preferidas no formato aceito pelo D´Accord Violão *Player* (formato dv3) aumentando assim o repertório disponível para usuários menos avançados e iniciantes.

O processo de edição de uma música não é, entretanto, uma tarefa trivial. Ele requer a manipulação de diversas mídias diferentes (texto, sons etc) com suas peculiaridades em relação à forma de aquisição e modificação. D´Accord Violão Editor lida com esse problema, dividindo o processo em passos, como pode ser visto na Figura 2.

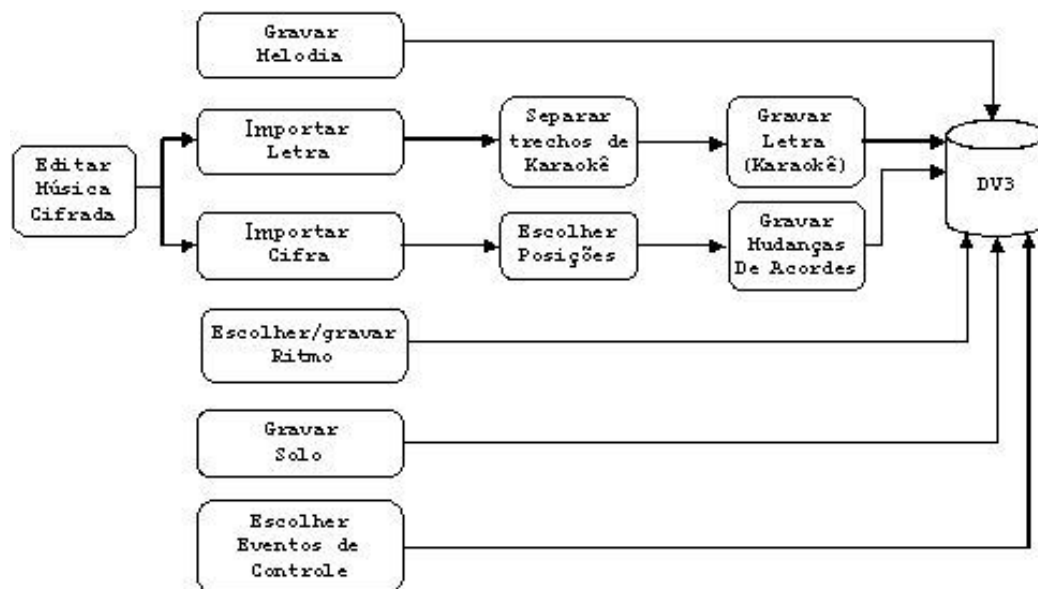


Figura 2 – Processo de gravação do D´Accord Violão Editor dividido em passos –  
Adaptado de (Cabral 2002)

Cada um dos passos está relacionado a um tipo diferente de mídia e eles podem ser feitos concorrentemente ou seja, não há uma sequência determinada a ser seguida.

Para gravação da melodia e do solo, insere-se o arquivo MIDI correspondente a cada um. O programa encarrega-se de gerar a melodia no formato proprietário D´Accord Violão.

A cifra e a letra são inseridas em modo texto. Para criar o letreiro de karaokê, o usuário deve sincronizar o tempo das notas da melodia e dos acordes na harmonia com o tempo do evento de seleção do texto e de seleção de acordes respectivamente, utilizando o teclado do computador.

Os eventos de controle (mudança de andamento e de volume do acompanhamento ou da melodia) são inseridos em uma lista que associa o tempo de ocorrência com o nome do evento e os parâmetros aceitos por ele.

## 2.4.Limitações da Edição de Ritmos no D´Accord Violão

O ritmo no D'Accord Violão pode ser visto como a aplicação de eventos de “ligar” (tocar) e “desligar” (para de tocar) as notas de um acorde seguindo um padrão de tempo bem definido, em um intervalo de tempo determinado. No momento em que há uma mudança de ritmo, há, efetivamente, uma mudança no padrão com que as notas são ligadas e desligadas. Esse conceito pode ser mais claramente observado na Figura 3.

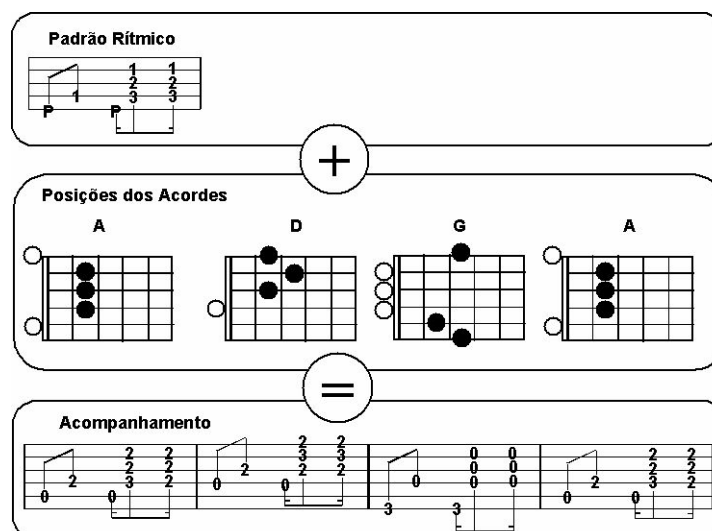


Figura 3 – Ilustração da aplicação do ritmo aos acordes – Extraído de (Cabral 2002)

É possível editar o ritmo de uma música de duas formas no D'Accord Violão: a primeira é escolhendo o padrão rítmico a ser usado durante toda sua música, a partir de uma base de padrões previamente construída, e a segunda é criando o ritmo a partir do zero, utilizando o próprio teclado do computador (digitando teclas específicas correspondentes aos eventos desejados, no momento correto).

O principal problema da primeira abordagem é que geralmente conduz a resultados musicalmente pobres, pois, se a base de ritmos contiver poucos padrões, é muito provável que o usuário não consiga a diversidade rítmica que deseja em sua música. No caso antagônico, se a base contém uma grande quantidade e diversidade de padrões, apesar da potencial qualidade do acompanhamento resultante, é muito mais difícil encontrar o padrão rítmico desejado.

O grande problema da segunda abordagem é que, embora possibilite alcançar resultados musicais extremamente satisfatórios, uma vez que propicia ao usuário total liberdade criativa, essa abordagem acaba sendo inviável devido à dificuldade e à demanda de tempo que a tarefa consome. No melhor caso (quando não houver necessidade de reedição), o tempo para gerar exclusivamente o ritmo de uma música é igual à sua duração. Na prática, porém, correções e reedições são geralmente necessárias, tornando a tarefa ainda mais demorada.

Além disso, os resultados práticos obtidos na geração do ritmo através do teclado do computador não são encorajadores. Problemas como atraso de tempo ou “desaparecimento” de notas ocorrem freqüentemente. Isso acontece devido principalmente ao fato de que esse dispositivo não foi projetado para desempenhar atividades que necessitem de respostas no tempo tão precisas.

Um violão MIDI (Hill 1994) poderia ajudar muito na geração no processo de aquisição de um acompanhamento rítmico personalizado. O problema, neste caso, é que nem todas as pessoas dispõem desse tipo instrumento, bastante caro e de difícil manipulação. Ajustes da intensidade de captação das cordas específicos para cada usuário do instrumento e a necessidade de controlar eventos como *pitch-bend* são alguns exemplos de tal dificuldade.

Outros instrumentos MIDI, principalmente teclados, são mais populares, baratos e de manipulação mais fácil. Porém, neste caso, a falta de naturalidade no ato da inserção do ritmo (trata-se de um ritmo de violão sendo simulado em um teclado) é bastante semelhante a do teclado do computador.



A geração automática de linhas<sup>1</sup> musicais através do computador, neste caso específico, o acompanhamento rítmico, poderia ser utilizada no contexto da edição de ritmos no D'Accord Violão, como uma alternativa para diminuir a demanda de esforço no referido processo de edição, grande problema encontrado na segunda abordagem, mantendo ainda qualidade nos resultados obtidos, problema encontrado na primeira alternativa.

A geração automática de acompanhamentos musicais é um problema que vem sendo estudado há muitos anos (Roads 1995) e, embora muitos avanços tenham sido feitos nesse caminho, não existe, hoje, uma solução definitiva para o problema, como será visto a seguir.

## **2.5.Sistemas de Acompanhamento Automático**

Os Sistemas de Acompanhamento Automático, ou SAAs, (Ramalho 1999) são programas capazes de gerar, em tempo real ou não, linhas de acompanhamento para diversos tipos de instrumento (linha melódica para baixo, linha rítmica para a bateria etc.). Eles simulam um músico humano em processo semelhante ao da improvisação.

São muito freqüentemente utilizados por arranjadores e músicos profissionais. Por arranjadores, para não precisarem se preocupar com o arranjo de todos os instrumentos (por exemplo, para criar o arranjo de guitarra, mas sem se preocupar com o arranjo do baixo e da bateria). Por músicos, para servir de assistente de ensaio (por exemplo, para gerar um acompanhamento que servirá de base para um solo).

De maneira semelhante à de um músico humano ao desempenhar a mesma tarefa, os SAAs dispõem de apenas um pequeno número de informações para guiá-los na criação da linha musical. São elas:

- O estilo musical ao qual pertence;

---

<sup>1</sup>Uma linha musical é qualquer segmento musical executado por um dado instrumento.

- a grade de acordes (ver Figura 4);
- aquilo que foi criado e tocado até o momento por ele próprio e
- a percepção das outras linhas musicais sendo executadas (no caso de existirem outros músicos, humanos ou não).

E m7(b5)	A 7		C m 7	F 7	
F m 7	Bb 7		Eb maj7	Ab 7	
Bb maj7	Em7(b5)	A7	D m7	G m7	C7
F maj7	G m7	C7	Am7(b5)	D 7	
G 7	G7		C m7	C m7	
Eb m7	Eb m7	Ab 7	Bb maj7	Bb maj7	
E m7(b5)	A 7		D m7(b5)	G 7	
C m7(b5)	F 7		Bb maj7	Bb maj7	

Figura 4 – Grade de acordes de “Stella by Starlight” (de N. Washington and V Young) –  
Extraído de (Ramalho 1997)

Evidentemente, há uma série de dificuldades em efetivamente criar uma linha musical a partir de um cenário tão restrito e limitado. Essas dificuldades são descritas a seguir.

### 2.5.1.O Problema da Formalização do Conhecimento Musical nos SAA

A falta de formalização do conhecimento é sem dúvida o principal problema a ser levado em consideração pelos SAAs. Este problema é, na realidade, pertinente a qualquer atividade onde haja necessidade de simular ou imitar o processo cognitivo humano no que se refere à sua capacidade criativa (Spector 1994). Apesar de todos os avanços científicos na área (Boden 1992), a criatividade humana continua sendo um grande enigma.

Especificamente sobre música, um músico humano geralmente é incapaz de explicar formalmente o motivo da sua decisão de tocar determinadas notas em detrimento de outras (Ramalho 1997). De fato, o músico parece não seguir um processo bem definido que o conduza até as notas a serem tocadas.

Como exemplo desse fato podemos ver, na Figura 5, trechos extraídos do repertório do grande trompetista de Jazz, Miles Davis. Apenas em um pequeno fragmento de dois compassos com contexto harmônico Dm7 – A7 é possível encontrar quatro tipos de padrão tocados pelo músico (Baker 1980). Nenhum deles contraria a harmonia, o ritmo ou o estilo daquilo que está sendo tocado.



Figura 5 – Quatro improvisações sobre Dm7| A7 encontradas em Miles Davis – Extraído de (Ramalho 1997)

É ainda mais difícil ponderar sobre uma espécie de “algoritmo universal”, seguido de forma unânime pelos músicos que, a partir das informações disponíveis (o estilo, a grade de acordes, o que foi tocado e o que está sendo tocado no momento), crie um acompanhamento novo.

De maneira distinta de procedimentos como, por exemplo, fazer uma adição de dois números ou trocar o pneu do carro, o processo de criação artística parece transitar por caminhos mais incertos; menos exatos (Ramalho 1994; Spector 1994).

Na realidade, o estilo, a grade de acordes e as outras informações disponíveis apenas dão leves sugestões sobre o que deve ser executado. Cada músico interpreta as informações da sua maneira.

A escolha das notas a serem tocadas não parece ser feita de forma explícita, e sim a partir de fatores subjetivos como, por exemplo, experiência auditiva anterior (lembrar de fragmentos musicais, dele próprio ou de outro músico, que já foram tocados em contextos semelhantes e adaptá-la ao contexto atual) e de uma enorme

quantidade de critérios como a intenção em aludir a sensações igualmente subjetivas e nebulosas, tais como tensão, dinamismo, *swing* etc. Esse conjunto de fatores, aliados evidentemente ao conhecimento musical, conduz o músico pelo tortuoso caminho da improvisação.

Adicionalmente, no caso em que a geração de uma linha musical faz parte de uma criação coletiva, o músico deve continuamente perceber o que os seus companheiros estão fazendo e adaptar-se a eles (Ramalho 1997). O fato de que esta interação ocorrer em tempo real só agrava a complexidade da tarefa.

Em resumo, a dificuldade e também o grande interesse no problema de acompanhamento automático estão justamente na modelagem do conhecimento utilizado pelos músicos para superar a enorme distância que separa as instruções incompletas da grade de acordes e as notas que efetivamente serão tocadas (Ramalho 1994).

É importante deixar claro que a modelagem do conhecimento dos músicos não é, evidentemente, a modelagem de como esse processo é executado na mente humana. Porém, para entender como a mente funciona, é necessário primeiro ter um bom conhecimento de que tipo de informação está em jogo (Johnson-Laird 1991).

### 2.5.2.Dificuldades na Abordagem de Agentes

Usando os conceitos da Inteligência Artificial, poderíamos pensar em um SAA como sendo um agente inteligente (Russell 1995). Seguindo o modelo *PAGE* (*Perceptions, Actions, Goals e Environment*) para definição de agentes, temos que suas percepções são a grande de acordes, o que foi tocado anteriormente, e o que os outros músicos estão tocando e sua principal ação é tocar.

Alem disso, podemos afirmar que a tarefa de gerar acompanhamento musical desenrola-se em um ambiente extremamente complexo. Tais ambientes caracterizam-se por serem **dinâmicos**, pois o ambiente continua mudando (novas notas estão sendo tocadas) enquanto o agente está deliberando (criado sua participação). Trata-se de um processo que exige respostas em tempo real. **Não-determinísticos**, pois o próximo estado do ambiente não pode ser determinado

apenas pelo estado atual e as ações já tomadas anteriormente (partes do acompanhamento já criadas). **Contínuos**, já que não há um número discreto de ações e percepções possíveis. **Não-episódicos**, pois a decisão sobre que notas devem ser tocadas depende fortemente do que foi tocado no passado. **Acessíveis**, uma vez que é possível adquirir todas as informações perceptivas contempladas pelo agente a qualquer momento. Porém, essa aparente simplicidade, esconde a dificuldade de como interpretar as informações perceptivas recebidas e associá-las a ações (Rowe 1993).

O problema maior está relacionado com o objetivo. Podemos deduzir da discussão da seção 2.5.1 que não é possível estabelecermos um objetivo bem definido, como, por exemplo, tocar nas notas dó, ré e fá em sequência. Podemos, em vez disso, traçar intenções pouco formais como, por exemplo, tocar bem, divertir-se etc.

Com isso, percebemos a dificuldade de formalizar a atividade de geração automática de acompanhamentos musicais como resolução de problemas (Newell 1972). Além da dificuldade em criar agentes que lidem bem com a complexidade do ambiente apresentado, os seus objetivos são mal definidos e dinâmicos (Pressing 1988; Ramalho 1994).

## **2.6.Decisões a serem tomadas pelos SAA**

Além de toda a dificuldade teórica já citada, é importante salientar as dificuldades práticas associadas ao processo de geração automática de acompanhamento. Qualquer um que deseje criar um SAA, e, portanto, receber como entrada do sistema o estilo, a grade de acordes, o que o sistema já tocou e o que outros músicos estão tocando, deve ter em mente a necessidade de analisar os seguintes critérios (Ramalho 1997):

**Qual a granularidade das decisões?** Como foi dito anteriormente, o ser humano nem sempre é capaz de explicar suas escolhas musicais nota por nota. Conclui-se, portanto, que ele não decide, na maioria dos casos, cada uma das notas que vão ser tocadas separadamente. Além disso, trata-se de um problema em tempo real, onde as respostas devem ser dadas quase instantaneamente. Faz-se necessário que

o problema da geração de acompanhamento de uma música seja dividido em um conjunto de subproblemas, cada um deles encarregado de gerar acompanhamento para uma “parte” específica da música. Dessa forma, é possível analisar e decidir o que vai ser tocado de forma menos reativa.

O tamanho de cada uma dessas “partes”, que podem ser chamados de grãos, deve ser cuidadosamente analisado. Esse fator está intimamente relacionado a duas características antagônicas entre si: a continuidade, isto é, a capacidade de criar um acompanhamento fluido, sem rispidez; e a velocidade de resposta, ou seja, o tempo necessário para o acompanhamento adaptar-se ao contexto (uma mudança repentina na melodia provocada por outro músico, por exemplo). Quanto maior o grão a ser usado, melhor será a continuidade do sistema, em contrapartida, tendem a reduzir a velocidade de resposta. Alguns grãos comumente usados são tempo, compasso e seqüências típicas de acordes (por exemplo, II–V–I, II–SubV–I), também chamadas de *chord chunks* (Ramalho 1997).

Ainda sob a análise das duas características supracitadas (continuidade e velocidade de resposta), é necessário também que se decida o seguinte: *O quanto do que foi tocado no passado deve ser levado em consideração na decisão atual?* Ou, em outras palavras: *qual a janela de tempo usada no sistema?* Quanto maior a janela, melhor será a continuidade do sistema, porém com menos velocidade de resposta. Além dessas duas características, é importante notar que a influência do passado deve ser analisada sob outras duas características. São elas: coerência, isto é, a capacidade de não se repetir freqüentemente e de manter uma espécie de “nexo musical”; e tempo de processamento, o tempo que se leva para analisar as informações disponíveis. Levar em consideração tudo que já foi tocado possibilita a maior coerência possível, porém, quanto mais perto do final da música, maior o tempo de processamento exigido.

As respostas a essas perguntas estão intimamente relacionadas ao processo denominado de *segmentação*. A segmentação pode ser vista como o particionamento do fluxo musical em unidades lógicas menores, a fim de facilitar o processo de análise musical (Trajano 2003). De fato, encontrar as respostas às referidas perguntas é encontrar uma segmentação adequada para a música. Porém, segmentar um fluxo musical não é uma tarefa simples, sobretudo por dois

motivos. O primeiro é o fato de que é possível segmentar uma música usando diversos critérios diferentes, portanto, uma determinada música pode ser dividida em segmentos muitas vezes diferentes. A segunda é que muitos dos critérios usados para segmentar estão relacionados à estética, à estilística e até mesmo a características culturais, muitas vezes não sendo critérios objetivos (Trajano 2003).

## **2.7. Peculiaridades do Acompanhamento Rítmico para o Violão**

Façamos um breve resumo do que foi dito até aqui: dada a grande dificuldade imposta aos usuários do D'Accord Violão Editor em gerar um acompanhamento rítmico de qualidade sem grande demanda de tempo, optou-se por uma abordagem de geração automática desse acompanhamento usando os conceitos dos Sistemas de Acompanhamento Automático. Temos, agora, que saber como encontrar as informações que são necessárias aos SAAs (estilo, grade de acordes, o que foi tocado anteriormente e o que os outros músicos estão tocando) no contexto do D'Accord Violão.

Primeiro, como o estilo da música deve ser conhecido pelo usuário no momento da edição, ele próprio será responsável por indicar essa informação. Segundo, a grade de acordes no D'Accord Violão é gerada no processo de edição da cifra (descrito na seção 2.3.2). Terceiro, o sistema tem total acesso ao que já foi tocado, este deve apenas armazenar essa informação para usá-la posteriormente. E quarto, embora não exista, objetivamente falando, nenhum outro músico no cenário do D'Accord Violão, podemos considerar a própria melodia da música como uma linha musical que influenciará na criação do acompanhamento rítmico.

Em relação aos problemas envolvendo o modelo geral de SAA descritos nas seções 2.5.1 e 2.5.2, a geração de acompanhamento rítmico para o violão apresenta algumas simplificações que merecem ser apontadas. A primeira é que não há necessidade de gerar o acompanhamento rítmico em tempo real, uma vez que esta faz parte de um processo de edição e que o resultado final deve ser validado pelo usuário. A segunda é o fato da melodia da música representar uma linha musical muito mais restrita que aquela produzida por músicos em tempo

real. Na realidade, a melodia não apresenta nenhuma imprevisibilidade durante a execução.

Em contraste a essa simplificação, há um agravamento do problema da formalização do conhecimento. Ao contrário de uma extensa literatura sobre harmonia e contraponto que auxilia sobremaneira na modelagem de acompanhamentos melódicos e harmônicos, quase não há literatura sobre técnicas de emprego de acompanhamento ritmo (Ramalho 1999). A hipótese que surge dessa escassez da literatura é que o acompanhamento rítmico, em contraste com outros tipos de acompanhamento, possui um componente de introspecção muito mais profundo e intimista. Essa afirmação baseia-se no comentário tecido por Schönberg, em seu Tratado de Harmonia (Schönberg 1922), sobre um possível princípio unificador entre a harmonia e o ritmo de uma música.

*Poder-se-ia, no máximo, tentar ordenar as incontáveis relações entre harmonia e ritmo de acordo com características bastante gerais. Eu duvido muito que a partir daí poderia surgir algum princípio unificador. Eu acredito que seria tão difícil quanto descobrir uma chave para todas as relações possíveis de luz e sombra de um determinado objeto, em todas as horas do dia e em todas as estações do ano, com todas as formas imagináveis de nuvens. E seria, de fato, uma tentativa supérflua. A maioria das regras rítmicas dos velhos livros são sempre restrições [...], nunca mostram como se deve combinar e sim como não se deve. E estas restrições têm quase que exclusivamente a função de evitar ou esconder certas dissonâncias (que para nós não são mais dissonâncias) (Schönberg 1922).*

A única ressalva que fazemos a esse comentário é que hoje, ao contrário da época em que o livro fora escrito (1922), podemos contar com o auxílio do computador para avaliação de grandes massas de dados.

Para o desenvolvimento de um SAA para acompanhamento rítmico faz-se necessário responder as seguintes questões principais.

- Quais são os padrões rítmicos usados em um dado estilo musical?
- Como ocorre a escolha de um fragmento específico em detrimento de todos os outros?
- Como ocorre a variação dos padrões em uma determinada música?
- A escolha dos padrões é influenciada pela melodia da música? Como ocorre essa influência?



- A escolha dos padrões é influenciada pela harmonia da música?  
Como ocorre essa influência?
- A escolha dos padrões é influenciada pelo andamento da música?  
Como ocorre essa influência?

São essas questões que tentaremos responder no decorrer dessa dissertação.

### 3. Estado da Arte

Neste capítulo, discutiremos diversas técnicas relacionadas à solução do problema de geração automática de linhas musicais, mais especificamente, de Sistemas de Acompanhamento Automático. Embora não seja uma lista exaustiva dos métodos e sistemas existentes, apresentam-se aqui aqueles que mais se destacam. É importante enfatizar que, do nosso conhecimento, não existe nenhum programa que efetivamente trate o problema de acompanhamento rítmico para o violão (embora o software *Band-in-a-box* trate da geração combinada de acompanhamento harmônico e rítmico para violão). Portanto, esse capítulo não se destina a fazer um comparativo entre sistemas equivalentes e seus prós e contras, e sim, a analisar aqueles que possuem em comum o fato de gerarem acompanhamento musical e partirem de pressupostos semelhantes para resolver o problema. Antes de iniciarmos a discussão, vale a pena fazer uma breve contextualização do problema sob a ótica da Computação Musical.

#### 3.1.Contexto do Problema

É bastante natural pensar que na Computação Musical existem tantas subáreas quanto existem os meios do ser humano interagir com a música (Roads 1995). Baseados nessa afirmação, podemos deduzir subáreas como, por exemplo, a performance e a expressividade, que estuda como fazer um computador executar (interpretar) uma música de modo semelhante ao feito por um ser humano (Widmer 1995; Widmer 2001); a análise musical por computador, que permite minimizar e, em alguns casos, simular o trabalho de um musicólogo em tarefas como, por exemplo, descoberta de padrões (Rolland 2001); e a composição automática. É nesta última onde está inserido o nosso trabalho. A Figura 6 traça um caminho a ser percorrido entre as subáreas da Computação Musical até chegar ao nível do trabalho proposto nessa dissertação.

A consequência imediata de uma análise desta figura é a percepção que a hierarquia das subáreas é também a hierarquia da complexidade dos problemas abordados nela. Deste modo, podemos ressaltar três pontos de fundamental importância.

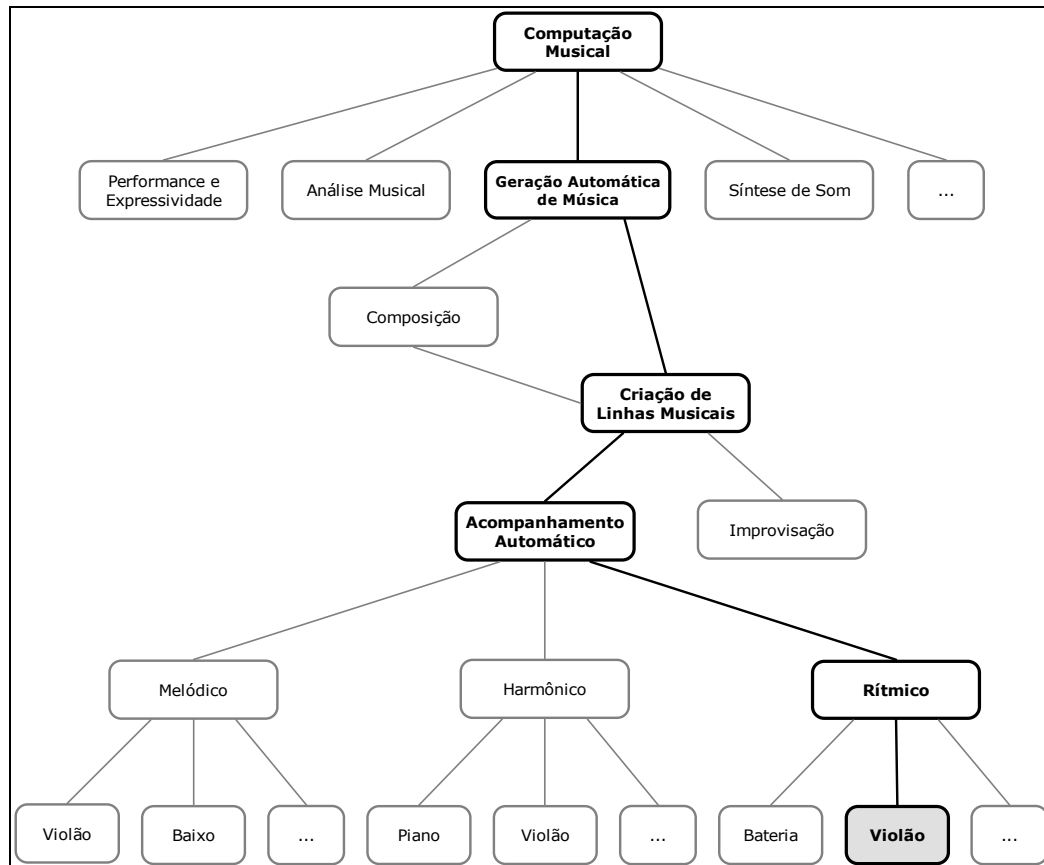


Figura 6 – Contextualização do problema de acompanhamento rítmico para violão em relação a Computação Musical

O primeiro refere-se ao fato de que há uma grande diferença de complexidade entre a geração de linhas musicais e a composição por computador. Enquanto na primeira há restrições, ainda que pequenas, no universo de notas a tocar pela necessidade de limitar o desenvolvimento da música ao da harmonia definida pelos acordes da grade, na segunda efetivamente não há restrições (a não ser aquelas definidas pelo próprio estilo musical escolhido). Neste último caso, a própria grade de acordes também deve ser criada pelo computador. É possível pensar no problema de geração de linhas musicais como um subproblema da composição. Uma vez decididas algumas características da música a ser composta, como o estilo, intenções, grade de acordes etc., é possível usar as técnicas de

geração de linhas musicais para criar efetivamente as linhas dos instrumentos escolhidos.

O segundo ponto trata da diferença entre acompanhamento e improvisação. Esta diferença não está na geração da linha musical em si, mas, na organização dessas linhas como resultado de uma criação coletiva. Em um conjunto musical, há uma espécie de ordem de importância, onde o improvisador é a figura principal, aquele que possui mais liberdade criativa. Os outros instrumentos, aqueles que fazem o acompanhamento, devem ao máximo “seguir-lo” melódica, rítmica e harmonicamente (Walker 1994; Hidaka 1995). No que se refere ao modo de gerar a linha musical, essa diferença exerce influência apenas sobre grau de liberdade em relação ao que está sendo tocado pelos outros. Por exemplo, na improvisação, os acompanhadores devem dar preferência a seguir o improvisador, mesmo quando o que este último toca contraria as indicações da grade de acordes. Por isso, mencionaremos sistemas de improvisação neste capítulo com a certeza de não incorremos em erro conceitual.

O terceiro ponto aborda o fato de encontrarmos o violão abaixo dos ramos dos instrumentos melódicos, harmônicos e rítmicos (ver Figura 6). Isso acontece porque o violão é flexível o suficiente para assumir diferentes papéis em uma música: pode ser solista, sendo, neste caso, um instrumento melódico, ou acompanhante, sendo um instrumento harmônico/rítmico. Neste último caso, como foi dito no capítulo anterior (seção 2.2), pode-se separar perfeitamente as atividades da mão esquerda, responsável pela parte harmônica, e das atividades da mão direita, responsável pelo ritmo.

Retornando a discussão, a fim de simplificar a exposição, separaremos os sistemas em dois grupos, de acordo com sua abordagem em lidar com o problema da formalização descrito no capítulo 2 (c.f. seção 2.5.1): o grupo daqueles que geram a linha musical nota por nota, e o grupo daqueles que reusam fragmentos musicais previamente construídos. Os sistemas serão analisados sob três aspectos principais: a tecnologia utilizada para resolver o problema, a granularidade das decisões, e a janela de tempo utilizada, os dois últimos discutidos no capítulo anterior (seção 2.6).

### 3.2. Abordagem nota por nota

Os sistemas utilizando a abordagem “nota por nota” podem ser divididos em duas vertentes: a dos programas que procuram diretrizes objetivas para a criação das linhas musicais, e aqueles que criam as linhas musicais novas através de transformações aplicadas a uma linha dada. Como representantes do primeiro grupo, discutiremos alguns programas que utilizam o formalismo de Gramáticas (seção 3.2.1) (Carroll 1989). Os representantes do segundo grupo serão os programas que utilizam métodos evolutivos (seção 3.2.2), mais especificamente Algoritmos Genéticos (Mitchell 1996; Mitchell 1997).

É importante enfatizar que os grupos possuem muito mais elementos do que se deixa transparecer aqui. Como exemplo, podemos citar como elemento do primeiro grupo o *CHORAL* (Ebcioglu 1992), um sistema dedutivo baseado em regras (350 regras) a criação das linhas de baixo, tenor e alto, dada uma linha de soprano, produzindo um coral no estilo de J. S. Bach. Já no segundo grupo, (Levitt 1983) utiliza *Constraint Satisfaction Programming* (Russell 1995) para transformar e refinar uma melodia dada de entrada, produzindo uma nova saída.

#### 3.2.1. Sistemas baseados em Gramáticas

Segundo (Steedman 1984), “A idéia de que existe uma gramática de música é provavelmente tão antiga quanto a idéia da própria gramática”. De fato, a literatura é bastante vasta nesse aspecto (Schurmann 1989). Especificamente em Computação Musical, em 1983, essa abordagem foi reforçada pelo lançamento do livro *The Generative Theory of Tonal Music* (Lerhdal 1983), que influenciou grande parte dos trabalhos na área.

Em seu clássico artigo (Johnson-Laird 1991), o autor sugere o uso de uma gramática regular para a geração do ritmo das notas tocadas pela melodia e, de uma gramática sensível ao contexto para a geração de uma grade de acordes ou substituição de acordes da grade. O autor também descreve um programa-teste capaz de gerar linhas de contrabaixo no estilo *Standard Jazz* a partir de uma grade de acordes. Ele define uma gramática regular relativa ao contorno melódico das notas. A Figura 7 apresenta o diagrama de transição de estados para a geração da

linha de contrabaixo. Os itens, **f**, **d**, **s** e **i** correspondem, respectivamente a primeira nota tocada, a repetição da nota anterior, a nota descendente e a nota ascendente, as últimas duas não necessariamente tons vizinhos. Por exemplo, uma nota descendente num contexto de C, após ter sido tocado uma nota sol, poderia ser um mi (não necessariamente um fá).

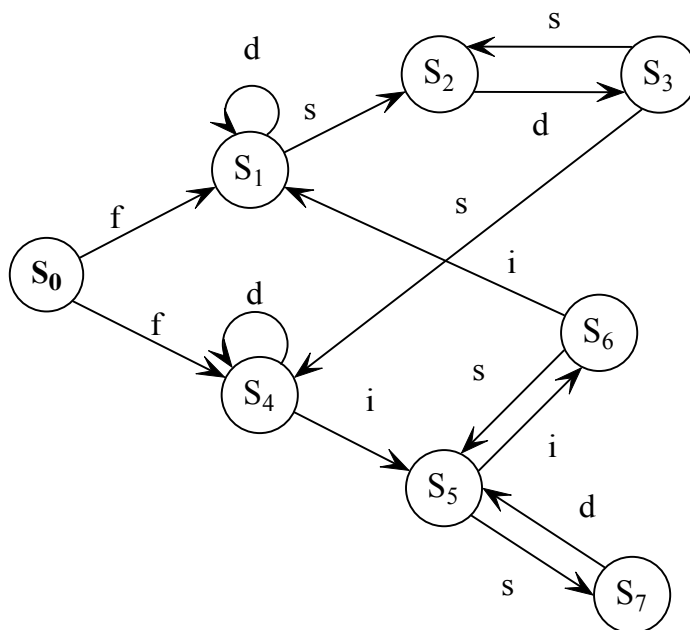


Figura 7 – Diagrama de transição de estados da gramática de geração da linha de contrabaixo

Para cada compasso, uma vez que o programa selecionou o contorno da próxima nota a ser tocada, deve-se escolher a altura da nota. Se o contorno escolhido é **d**, ou seja, repetição da nota anterior (janela de tempo de uma nota), sua altura já está definida, caso contrário o contorno apenas restringe quais as notas que podem ser tocadas, com o auxílio de restrições harmônicas que indicam quais notas podem ser tocadas dado o acorde atual da grade. Por exemplo, com o acorde C7, além das notas do acorde (C,E,G e B), é possível que o contrabaixo toque as seguintes notas de passagem: D, F# e A.

Restrições harmônicas são aplicadas no início de um novo acorde e no tempo dos compassos, assim, notas de passagem são evitadas quando um novo acorde inicia ou no primeiro tempo do compasso.

É importante citar que o autor não leva em consideração o aspecto rítmico da linha de baixo. Todas as notas possuem a mesma duração.

Abordagens semelhantes podem ser encontradas em diversos trabalhos. O *Cybernetic Composer* (Ames 1992) e o trabalho de Giomi e Ligabue (Giomi 1991) utilizam a mesma tática de geração das linhas musicais. O primeiro gera linhas de piano, contrabaixo e bateria em um dos quatro estilos musicais disponíveis (Rock, *Standard Jazz*, *Latin Jazz* e Ragtime), enquanto o segundo gera uma linha de saxofone em *Standard Jazz*. Em ambos os programas, o primeiro passo é definir, para cada compasso, o momento e a duração das notas, para cada instrumento. Utiliza-se, para isso, uma gramática cujas regras são associadas à probabilidade, formando uma estrutura chamada árvore probabilística derivacional (APD), como exemplifica a Figura 8. Uma outra APD é utilizada para escolher a altura das notas previamente estabelecidas.

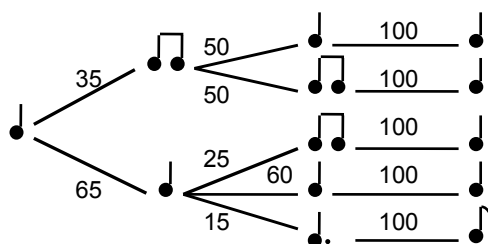


Figura 8 – Exemplo de árvore probabilística derivacional para ritmo (pesos em %) –  
Extraído de (Ramalho 1997)

### 3.2.2.Algoritmos Genéticos

Os Algoritmos Genéticos são bastante utilizados na geração automática de linhas musicais, uma vez que fornecem uma alternativa poderosa e eficiente para lidar com problemas cujo espaço de soluções é bastante amplo (que é o caso da geração automática de linhas musicais). Desta forma, é possível explorar soluções não triviais e até inesperadas, característica que pode ser desejável em sistemas que buscam resultados criativos (Papadopoulos 1999).

Em (Spector 1994), discute-se um sistema que produz linhas melodias de saxofone no estilo Bebop Jazz em um formato conhecido com *trading fours*, uma

espécie de jogo em que os instrumentistas se revezam a cada quatro compassos, improvisando sobre o tema criado pelo músico que o antecedeu. O sistema possui um grande conjunto de operadores de transformação, como por exemplo: REP, que recebe uma melodia de entrada e retorna uma nova melodia consistindo do primeiro compasso da melodia dada repetido quatro vezes; 8VA, que transpõe a melodia uma oitava acima, EXTEND, que preenche as pausas existentes de uma melodia com a própria melodia etc.

A cada geração (de novas “espécies”), a melodia de entrada é transformada utilizando os operadores. O resultado obtido é confrontado com fragmentos de quatro compassos previamente armazenados em uma base de casos de “qualidade assegurada” (os autores usaram cinco fragmentos de Charlie Parker). Uma função de aptidão (*fitness*) estabelece como vencedora a espécie que estiver em maior conformidade com a base de casos utilizando critérios como: coerência rítmica, equilíbrio tonal etc.

*GenJam*, ou *Genetic Jammer* (Biles 1994), é um programa evolutivo que produz frases *trade fours* a partir de uma frase anterior tocada por um ser humano em tempo real. O *GenJam* ouve o que o ser humano toca (interpretando a entrada MIDI), mapeia a performance deste na sua estrutura de cromossomos e responde ao usuário com o resultado da mutação obtida do que ele próprio tocou. Em outras palavras, o programa usa seus operadores de mutação (que fazem sentido musicalmente, como por exemplo, inversão, retrocesso, transposição) para transformar o que o músico humano tocou no que ele próprio irá responder. Percebe-se portanto que o *GenJam*, não usa, em nenhum momento, uma função de aptidão. Na realidade, segundo comentários do autor, o desempenho do seu programa depende sobremaneira do desempenho do músico que irá “dialogar” com ele.

### 3.3.Reuso de Fragmentos Musicais

Como foi dito no Capítulo 2 (seção 2.5.1), embora músicos profissionais usem regras que aprenderam nas escolas para criar acompanhamentos e improvisações, estas regras não incorporam todo o conhecimento que eles empregam ao tocar. De fato, não há um encadeamento de regras lógicas que mapeie diretamente



propriedades musicais de bastante relevância, tais como tensão, articulação e *swing*, em notas musicais efetivas. Especialmente em alguns estilos, como o Jazz, por exemplo, os músicos aprendem a tocar escutando e imitando performances de músicos famosos (Ramalho 1994; Walker 1994; Sabatella 1996).

Através deste processo empírico de aprendizado, os músicos adquirem um conjunto de exemplos de frases musicais que podem ser reusados em uma performance futura. Esta declaração é corroborada pela existência de diversos trabalhos musicológicos que objetivam a identificação de fragmentos típicos usados por músicos famosos, como Charlie Parker (Owens 1974) e Miles Davis (Baker 1980).

Neste momento, é necessário fazer uma observação. Geralmente, a literatura emprega o termo “padrão” em um sentido vago, incluindo todo fragmento musical que pode ser encontrado em um determinado *corpus*. Entretanto, a palavra “padrão” deve designar somente estruturas musicais recorrentes, isto é, fragmentos que ocorrem bastante frequentemente, de acordo com algum limiar específico (Rolland 1996; Rolland 2001). Neste ponto do texto, preferimos usar o termo “fragmento” devido à sua generalidade: todos os padrões são fragmentos, mas a recíproca nem sempre é verdadeira.

Do ponto de vista da geração automática de música por computador, é importante notar que a reutilização de fragmentos musicais é uma técnica extremamente poderosa para minimizar o problema da falta de formalização do conhecimento, citado na seção 2.5.1. Primeiramente, esses fragmentos podem facilmente ser adquiridos por consultas a peritos ou à literatura; ou usando programas da extração dos padrões musicais (Rowe 1993; Rolland 2001; Santana 2003). Em segundo lugar, os fragmentos representam o conhecimento em extensão, uma vez que codificam implicitamente o conhecimento nos exemplos de soluções concretas (fragmento apropriado) para um problema musical dado (uma sequência de acordes e o contexto associado).

A principal idéia por trás do reuso de fragmentos é recuperar e adaptar fragmentos musicais previamente armazenados e encadeá-los a fim compor novas linhas melódicas, harmônicas ou rítmicas. Isto não significa que improvisar e

acompanhar músicas é uma tarefa que se limita a encadear fragmentos previamente obtidos. Se assim o fosse, estaríamos fadados a ouvir improvisações e acompanhamentos de diversos músicos diferentes com pouca ou nenhuma distinção entre o material musical produzido. Não obstante, a forma simples e poderosa de atacar o problema da formalização do conhecimento provida pela abordagem de reuso de fragmentos abre caminho para a construção de sistemas de improvisação e de acompanhamento de resultados bastante satisfatórios (Ramalho 1997).

Além dos critérios de análise definidos na seção 2.6 (granularidade das decisões e a janela de tempo a ser levada em consideração), qualquer programa baseado em reuso de fragmentos deve levar em consideração uma série de critérios adicionais que definem seu comportamento e, em última instância, o seu desempenho. Esses critérios estão relacionados com a estrutura dos fragmentos a serem usados e com a forma de indexar e recuperar os fragmentos armazenados.

O primeiro critério trata de escolher se os fragmentos devem ser de tamanho fixo ou variável. Essa diferença reflete na facilidade de encadeamento dos fragmentos em proporção inversa à plausibilidade da música resultante. Embora usar fragmentos de tamanho fixo facilite o processo de encadeamento dos fragmentos, uma vez que existem lugares pré-determinados para encaixá-los, nem sempre os fragmentos encontrados em músicas são todos do mesmo tamanho. Na realidade, utilizar fragmentos do mesmo tamanho é mais uma comodidade que um fato musical. É importante notar que o critério de granularidade dos fragmentos está intimamente ligado ao critério do tamanho. Por exemplo, em determinado sistema onde se usa granularidade de compasso, é necessário que os fragmentos sejam de tamanho fixo. Já no caso do uso de granularidade de *chord chunks*, é necessário que os fragmentos sejam de tamanho variável.

O segundo critério aborda que tipo de informação é necessário para fazer a descrição dos fragmentos. Há dois grupos distintos de informação (Ramalho 1994). O primeiro grupo que reúne as informações relacionadas com o contexto musical onde o fragmento deve ser inserido. São exemplos de atributos de contexto: ritmo harmônico, posição relativa na música (introdução, refrão etc.), densidade da melodia e andamento. O outro agrupa as propriedades musicais que

o próprio fragmento contém, como, por exemplo, densidade, síncope e dissonância. Quanto maior o número de atributos de descrição utilizados, mais precisas e refinadas serão as escolhas, porém, com dois importantes efeitos colaterais. O primeiro refere-se à demanda de tempo no momento de recuperar o fragmento: quanto maior a quantidade dos atributos, mais tempo será consumido no processo. O segundo efeito colateral é o fato de que uma grande diversidade de atributos freqüentemente está relacionada ao uso intensivo de conhecimento específico do contexto de onde o fragmento foi extraído (por exemplo, utilizar atributos que são unicamente encontrados em determinados estilos musicais). Isso interfere de forma negativa na generalização da solução e também restringe a inserção de novos elementos no repositório de fragmentos.

O terceiro critério refere-se à escolha da técnica de recuperação dos fragmentos. Essa escolha deve levar em consideração o compromisso entre eficiência da recuperação e flexibilidade dos resultados (Ramalho 1994). Técnicas poderosas com uso de medidas de similaridade são bastante satisfatórias quanto à flexibilidade dos resultados (principalmente quando a descrição dos fragmentos é muito rica), porém não possuem grande eficiência, principalmente quando o repositório de fragmentos é muito grande. No outro extremo, uma escolha aleatória é extremamente eficiente, independentemente do tipo de descrição dos fragmentos, porém, nada pode ser dito quanto aos resultados obtidos.

O quarto critério trata de como modificar os fragmentos em busca de um melhor casamento com o contexto de reutilização, já que nem sempre o fragmento recuperado no repositório adequa-se perfeitamente às necessidades do novo contexto. Algumas adaptações como, por exemplo, transposição das notas e mudança de andamento são bastante simples de implementar; outras, porém, são bastante complexas, visto que muitas das características dos fragmentos são interdependentes. Por exemplo, modificar o contorno melódico (por exemplo, de descendente para ascendente) de um fragmento, gera a necessidade de modificar características harmônicas (Ramalho 1999).

Nas seções que se seguem, discutiremos alguns dos principais programas que utilizam o reuso de padrões para a geração automática de linhas musicais. Essa

discussão será feita com base nos pontos de análise discutidos até o momento, são eles:

- Granularidade das decisões;
- janela de tempo;
- técnica de aquisição das informações do ambiente;
- tamanho dos fragmentos (fixo ou variável)
- atributos de descrição dos fragmentos (de contexto e de propriedades musicais);
- técnica de recuperação usada;
- técnicas de adaptação disponíveis.

### **3.3.1. *Band-in-a-Box***

*Band-in-a-Box*<sup>2</sup> (BiB) é uma ferramenta comercial bastante popular para criação automática de acompanhamento e improvisação para violão, piano, baixo, bateria e conjunto de cordas (ver Figura 9). Esse programa possui, como principal abstração, o estilo musical. Essa abstração, que corresponde a um arquivo em formato proprietário BiB, reúne todas as informações necessárias para que seja possível gerar uma linha musical no estilo desejado, como por exemplo, andamento padrão do estilo, sua assinatura de tempo (3/4, 4/4 etc.), o volume padrão de cada um de seus instrumentos e, principalmente, os fragmentos musicais que podem ser usados por cada um dos instrumentos de acompanhamento.

Os fragmentos estão separados em categorias, chamadas subestilos, com o intuito de prover um meio de organizá-los quanto ao contexto onde devem ser incluído.

---

<sup>2</sup> *Band-in-a-Box* (versão 2004) é marca registrada da *PG Music Inc.* (<http://www.pgmusic.com>)

Todas as informações foram retiradas do manual do usuário do programa.

Por exemplo, há um subestilo chamado de finalização, no qual devem ser armazenados fragmentos que devem ser aplicados no fim da música. Cada subestilo pode armazenar trinta fragmentos de mesma duração, porém, para os instrumentos harmônicos e melódicos, há subestilos que armazenam padrões de duração de dois compassos, um compasso, dois tempos ou um tempo, sendo possível armazenar até 270 fragmentos para cada instrumento. No caso da bateria (instrumento rítmico) os subestilos permitem apenas fragmentos de um compasso, excetuando o subestilo de finalização, que deve possuir dois compassos para qualquer dos instrumentos, totalizando um máximo de 120 fragmentos. Portanto, um estilo BiB pode conter até 1620 fragmentos musicais diferentes.

Os fragmentos podem ser inseridos utilizando um instrumento MIDI ou através da edição de uma partitura disponibilizada pela interface gráfica do BiB. Cada fragmento deve ser configurado de acordo com seis atributos relacionados com o contexto onde ele deve ser aplicado. O primeiro e único atributo obrigatório é o *peso relativo*. Esse peso determina com que *frequência* o fragmento será utilizado em relação aos outros fragmentos do subestilo. Os valores podem variar de 1 (raramente utilizado) até 9 (utilizado todas as vezes). Dessa forma, um fragmento assinalado com peso 4 tem duas vezes mais chances de ser utilizado em uma música que um fragmento com peso igual a 2. O segundo é a *máscara de compasso*, um atributo opcional que determina em que compassos da música o fragmento pode ser aplicado. Alguns de seus possíveis valores são: “tocado em compassos ímpares”, “tocado no primeiro compasso antes de viradas” etc. O terceiro atributo, também opcional, é a *máscara de tempo*, que determina em que tempo do compasso o fragmento deve ser tocado. Pode assumir valores de 1 a 4 para cada um dos tempos de um compasso 4/4. O quarto é a máscara de grau de acordes, que associa o uso do fragmento a um determinado grau de acordes da escala estabelecida na música (por exemplo, I, IV, V7 etc.). O quinto é o *intervalo para o próximo acorde*, que restringe o uso do fragmento ao intervalo entre duas notas consecutivas. Alguns dos valores possíveis são: “todos os intervalos”, “quinta abaixo”, “segunda acima”, etc.

O processo de escolha dos fragmentos é bastante simples. A cada mudança de acorde, para cada instrumento, o programa extrai informações relevantes sobre o contexto da música naquele momento (se está no final da música, se é um

momento propício para viradas etc.). De posse dessas informações, há uma restrição no que se refere ao subestilo de onde o fragmento será recuperado. Por exemplo, no contexto de finalização, apenas os fragmentos contidos no subestilo de finalização poderão ser recuperados. Ainda sobre o contexto, há também as restrições relativas aos atributos associados a cada fragmento: aqueles cujos atributos não são desejáveis no contexto corrente não são utilizados. O BiB faz um sorteio aleatório respeitando a probabilidade de utilização de cada fragmento. Depois de recuperado, o fragmento é adaptado sob os aspectos de andamento e transposição. É importante notar que nenhuma informação sobre o que já foi tocado e sobre o que os outros músicos estão tocando são levadas em consideração nesse software.

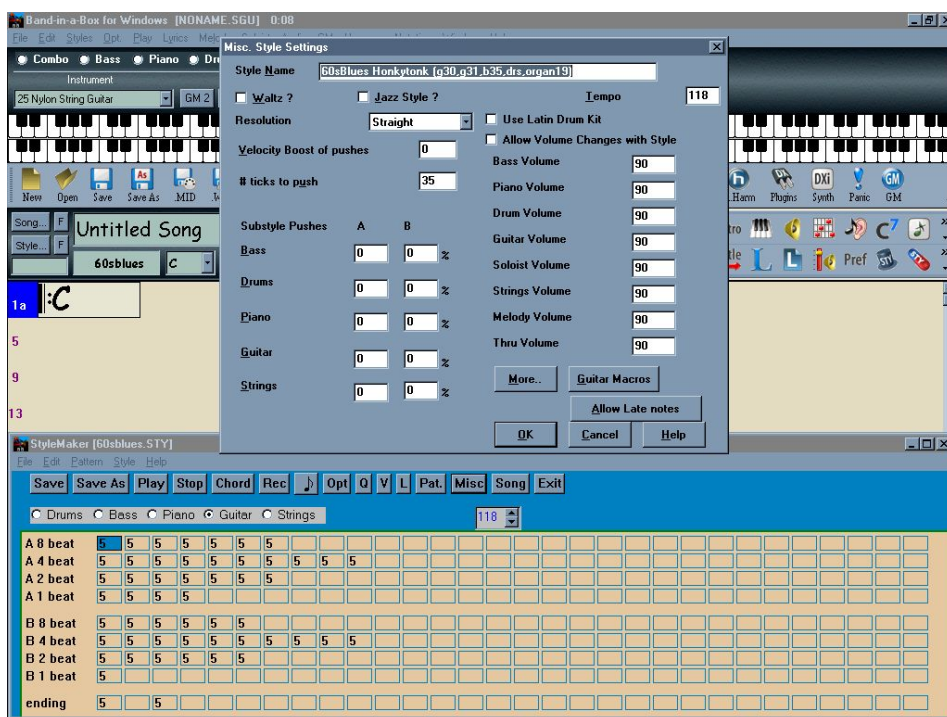


Figura 9 – Edição de Estilo no *Band-in-a-Box*

### 3.3.2.ImPact

Esse sistema foi desenvolvido pelo professor Geber Ramalho em seu doutorado (Ramalho 1997). Trata-se de um Sistema de Acompanhamento Automático que simula um contrabaixista em uma banda de Jazz numa performance ao vivo para um platéia.

O ImPact lida com os as dificuldades associadas aos SAA a partir de uma maneira particular de encarar o problema sob a ótica de agentes inteligentes. A Figura 10 ilustra a estrutura arquitetural do ImPact.

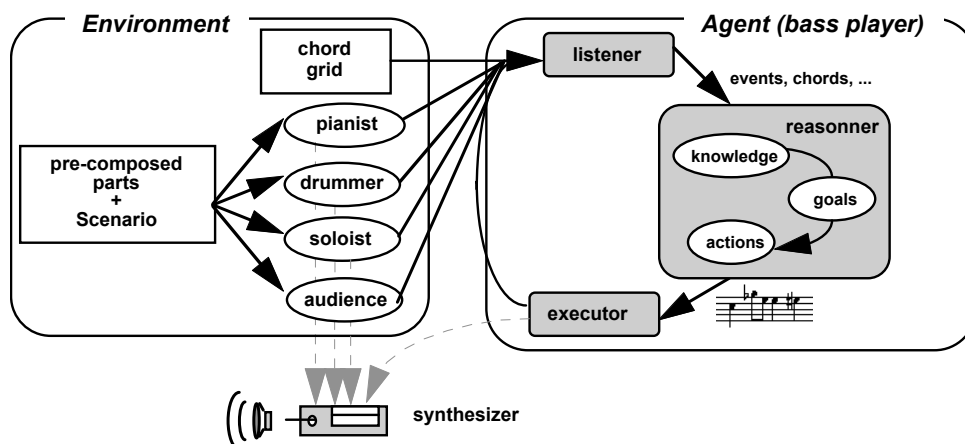


Figura 10 – Ilustração do ImPact como agente inteligente

Esta arquitetura reflete a divisão das tarefas do agente em percepção, raciocínio (ou planejamento) e ação. O lado direito da Figura 10 apresenta o agente contrabaixista, que é composto de três agentes especializados: o ouvinte (*listener*), responsável por captar as percepções, o raciocinador (*reasoner*), que planeja as notas a serem tocadas, e o executor (*executor*), que efetivamente toca as notas em seu tempo apropriado. Cada um deles será detalhado a seguir.

O ouvinte responde pela tarefa de perceber as informações relevantes do ambiente e “traduzi-las” para o raciocinador (fornecendo percepções mais objetivas). O lado esquerdo da Figura 10 mostra o ambiente contendo a grade de acordes, outros agentes músicos (pianista, baterista e o solista, comumente tocando algum instrumento de sopro, como saxofone) e um agente platéia. Cada um desses agentes é responsável por fornecer as informações que ele produz ao ambiente, ou seja, os agentes músicos fornecem o que estão tocando, enquanto o agente público fornece as reações do público em relação à performance do grupo.

A fim de evitar enfrentar os problemas de perceber todas essas informações em tempo real, o sistema usa a noção de cenário (Baggi 1992; Rowe 1993; Hidaka 1995). O cenário é uma representação simplificada do ambiente onde o agente está imerso. Ele é composto por dois tipos de eventos: os eventos produzidos

pelos outros músicos (por exemplo, “o baterista está tocando cada vez mais alto”) e os eventos produzidos pela platéia (por exemplo, “aplausos”, “público retira-se” etc.). O papel dos agentes músicos e público consiste em, continuamente, extrair do cenário sua ação específica e disponibiliza-la ao ambiente. Com isso, é possível garantir a simplicidade das percepções do agente contrabaixista e ainda manter uma coerência conceitual. Por exemplo, com o uso de cenários é possível simular o fato de um músico não poder ouvir o que ainda não foi tocado.

O executor simplesmente percebe a descrição das notas (altura, duração, momento inicial etc) planejadas pelo agente raciocinador, armazena essas informações e toca cada um das notas em um sintetizador MIDI no momento apropriado.

Antes de iniciarmos a discussão sobre como ImPact aplica o reuso de fragmentos, é necessário que façamos uma breve exposição sobre raciocínio baseado em casos, técnica fortemente utilizada no sistema.

### **3.3.2.1. Raciocínio Baseado em Casos**

Raciocínio Baseado em Casos, ou RBC, (do inglês *Case-Based Reasoning* (Kolodner 1993; Aamodt 1994; Mitchell 1997)) é um paradigma de resolução de problemas que, em muitos aspectos, é fundamentalmente diferente da maioria das abordagens de Inteligência Artificial (Aamodt 1994). No lugar de contar apenas com o conhecimento geral do domínio do problema, o RBC é capaz de usar um conhecimento específico previamente experimentado.

A principal idéia acerca deste paradigma é descrever e acumular descrições de episódios na área do conhecimento especializado e tentar descobrir, por analogia, quando um determinado problema é similar a um outro já resolvido. Desta forma, a solução já aplicada a um problema anterior pode ser utilizada ou adaptada para uma nova necessidade (Kolodner 1993).

Grande parte da importância e aceitação do RBC está na sua plausibilidade psicológica. Diversos estudos na área de psicologia cognitiva, destacadamente o de Roger Schank (Schank 1982), um dos principais idealizadores desse paradigma, dão evidências empíricas de que, em vários cenários, os seres



humanos utilizam seu conhecimento sobre problemas já solucionados para resolver outros semelhantes (Kolodner 1993; Strube 1995).

Um “caso”, na terminologia RBC, é uma descrição completa de um problema do domínio com a respectiva solução aplicada, além de, opcionalmente, uma avaliação da eficácia dessa solução. É descrito por atributos e seus respectivos valores. O conjunto dos casos é indexado por seus atributos mais significantes de forma a agilizar a busca e recuperação de casos similares. Uma das grandes vantagens da utilização do RBC é a forma simples de adquirir o conhecimento: o armazenamento do caso em si constitui a aquisição de conhecimento<sup>3</sup> (Mitchell 1997).

As principais tarefas a serem executadas por um sistema RBC são: identificar a situação-problema atual, encontrar um caso anterior com características similares ao problema em questão, usar o caso para sugerir uma solução ao problema, avaliar a solução proposta e atualizar o sistema para aprender com essa nova experiência. Cada uma dessas partes compõe o chamado ciclo RBC, também chamado ciclo dos quatro *REs* (Aamodt 1994) que pode ser visto na Figura 11.

---

<sup>3</sup> Isso não exclui, evidentemente, a necessidade de algum conhecimento mais geral no momento da descrição e manipulação dos casos (Strube 1995)

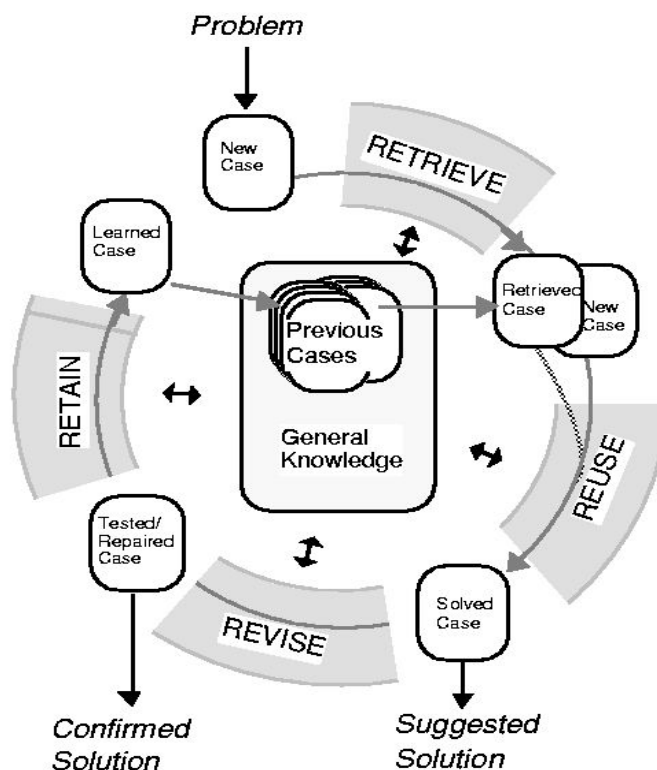


Figura 11 – Funcionamento do ciclo RBC, os quatro *REs*

No passo *RETRIEVE* (que chamaremos de recuperação), há uma consulta a um repositório previamente construído (base de casos) em busca de casos que tenham características similares às do problema a ser solucionado. Encontrados esses casos, o passo *REUSE* (reuso) adapta-os ao problema, focando em dois aspectos: as diferenças entre os casos recuperados e o novo caso, e quais partes dos casos antigos podem ser totalmente reaproveitadas no novo. No passo *REVISE* (revisão), avalia-se a solução gerada anteriormente e, no caso de falha, conserta-se a solução usando conhecimento específico do domínio. No passo *RETAIN* (armazenamento), qualquer conhecimento útil adquirido no processo é incorporado ao sistema. Esse conhecimento pode ser de sucesso, quando uma solução satisfatória é encontrada, ou de falha, caso contrário.

O desenvolvimento de um sistema baseado em casos envolve responder a seis perguntas básicas:

**Qual a natureza e conteúdo dos casos?** É necessário que os casos adquiridos sejam representativos, confiáveis e não conflitantes entre si (Kolodner 1993). Além disso, a granularidade dos casos deve ser analisada com bastante cuidado.

Por exemplo, um projeto arquitetônico de uma casa contém as necessidades de um cliente (o dono da casa) e a solução proposta através dos detalhes do projeto. Porém, considerar todo o projeto como um caso não seria de grande utilidade, já que dificilmente dois clientes teriam as mesmas necessidades em relação a uma casa inteira (Abel 2003).

**Como representar os casos?** O caso deve ser representado de forma tal que possível de ser entendido por um ser humano e que seja eficiente para a máquina. Algumas das muitas representações possíveis (cf. (Kolodner 1993) para uma lista mais ampla) são: vetores de características, associações atributo-valor (*frames*, redes semânticas, objetos), lógica de primeira ordem etc.

**Como indexar os casos?** A indexação consiste em estabelecer meios de recuperar os casos, adequada e rapidamente, no momento em que for necessário. Em geral, os índices são um conjunto de atributos que refletem bem as características relevantes do caso (Kolodner 1993), esses atributos podem ser escolhidos manualmente, a partir de um conhecimento sobre o domínio, ou a partir de métodos automáticos, como métodos indutivos (regras *ID3*, por exemplo).

**Como estruturar (organizar) os casos da base?** A forma como os casos estão organizados na base de casos influencia decisivamente na capacidade de recuperar o caso mais similar a um determinado problema. Há um compromisso entre acessibilidade (todos os casos podem ser acessados a cada momento) e eficiência (velocidade de recuperar o caso desejado). Existem várias estratégias de organização de casos na base de casos, duas das mais usadas são: memória plana (Figura 12), onde os casos são armazenados em forma de lista, possui alta acessibilidade porém baixa eficiência e rede de atributos compartilhados (Figura 13), estrutura hierárquica onde os valores dos atributos criam agrupamentos. (eficiência alta, mas acessibilidade baixa).

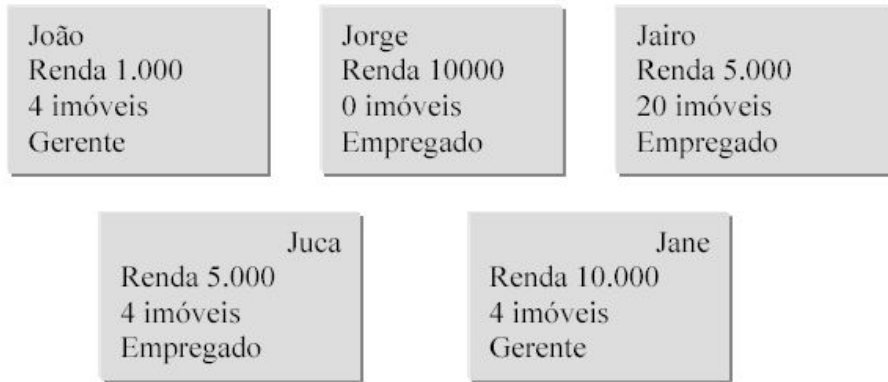


Figura 12 – Exemplo de Memória Plana

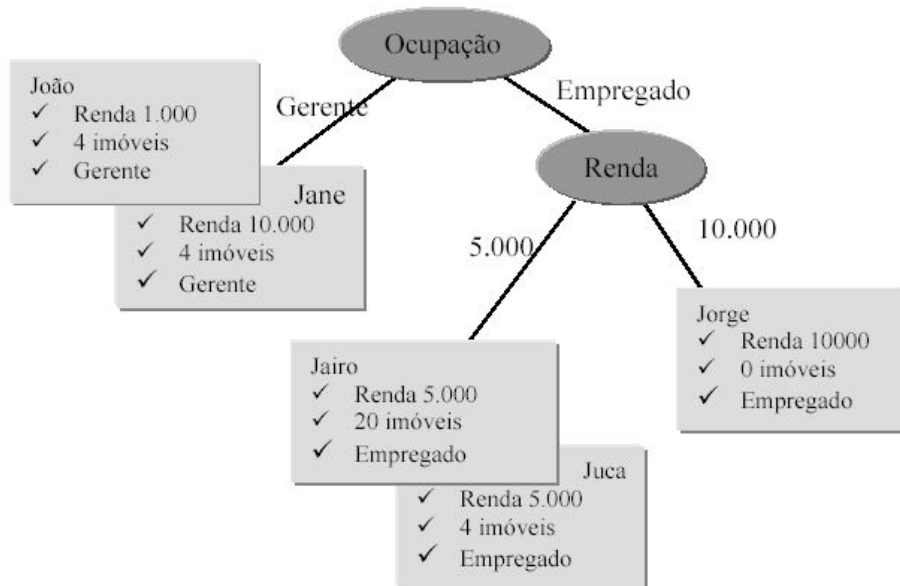


Figura 13 – Exemplo de rede de atributos compartilhados

### Quais são os critérios para a escolha do melhor caso e como recuperá-lo?

Dada uma descrição de um problema, uma estratégia de recuperação deve encontrar os casos mais similares à situação atual utilizando os índices da memória de casos. As estratégias utilizam os índices e a organização de memória para conduzir a busca dos casos potencialmente úteis. As mais simples fazem um exame exaustivo dos descritores (atributos com seus respectivos valores) dos casos e recuperam aquele com o maior número de descritores idênticos. Outras estratégias mais sofisticadas buscam o melhor caso, utilizando-se de heurísticas para reduzir o espaço de opções e direcionar a busca. Entre esses, pode-se citar:

busca serial, busca hierárquica e busca paralela (Kolodner 1993). Uma opção à busca por descritores idênticos é usar algum algoritmo de casamento (*matching*) parcial. Entre eles, o mais habitualmente usado é o *k-nearest neighbors* (*k-nn*) (Mitchell 1997). Esse método baseia-se na comparação entre um novo caso e aqueles armazenados na base utilizando uma soma dos valores de seus atributos ponderada por sua relevância nos critérios de similaridade. Uma possível função de similaridade é mostrada na equação abaixo.

$$Simil(X, Y) = \frac{\sum_{i=1}^n w_i \times sim_i(a_{xi}, a_{yi})}{\sum_{i=1}^n w_i}$$

Onde  $X$  e  $Y$  são casos com  $n$  atributos,  $w_i$  é o peso associado ao atributo  $i$ ,  $sim_i$  é a função de similaridade do atributo  $i$ ,  $a_{xi}$  e  $a_{yi}$  são os valores do atributo  $i$  para os casos  $X$  e  $Y$ , respectivamente. O resultado final da *k-nn* seria os  $k$  casos da base com maior valor na função de similaridade.

**Como adaptar o caso recuperado?** Normalmente, o caso selecionado não casa perfeitamente com a descrição do problema do caso desejado. Existem diferenças entre o problema existente e o caso contido na base que devem ser levadas em conta. O processo de adaptação procura por diferenças acentuadas entre as duas descrições e aplica regras de forma a compensá-las.

Na realidade, não existem regras gerais a serem usadas na adaptação (Abel 2003). Por exemplo, em um sistema de RBC para planejamento, a adaptação precisa reconhecer as pré-condições das etapas que devem ser atingidas e achar os planos para atingi-las. Já no caso de um sistema de diagnóstico, é necessário achar falhas nas explicações fornecidas e encontrar as causas não-registradas. Segundo Kolodner em (Kolodner 1993), as adaptações podem ser agrupadas em, métodos de substituição, onde os valores incorretos da solução são trocados por valores apropriados. Estão entre esses métodos a re-instanciação, usada no domínio de objetos (por exemplo, trocar frango por carne no contexto gastronomia), Ajuste de parâmetros no domínio numérico (por exemplo, recalcular o valor de um imóvel), métodos de transformação, onde há uma alteração estrutural da solução não aceita

em uma que funcione. Entre os métodos de transformação estão, as transformações de senso comum, onde há a aplicação de operadores simples para adicionar, apagar ou trocar elementos da solução.

Podem-se encontrar sistemas utilizando os conceitos de RBC em muitas áreas, como culinária, arquitetura, medicina, executando diversos tipos de tarefa como, preparar pratos, elaborar projetos, diagnosticar doenças etc. Na computação musical esse paradigma tem se mostrado bastante eficaz para resolver diversos tipos de problemas. É possível encontrá-lo em sistemas de harmonização de melodias, como GYMEL (Sabater 1998), que usa o conhecimento prático adquirido em casos anteriores para harmonizar uma nova música dada, em performance e expressividade, como o SaxEx (Arcos 1998), que transforma uma melodia tocada em um saxofone de modo inexpressivo em uma melodia expressiva. O sistema encontra, na base de casos, a melodia cujo contexto mais se adequa ao encontrado na melodia inexpressiva e aplica os mesmos operadores de expressividade (dinâmica, articulação, rubato etc.) à esta última.

Há também sistemas de composição automática (Pereira 1997) e de acompanhamento automático (Ramalho 1997) que usam raciocínio baseado em casos no processo de criação musical. De fato, esta é uma abordagem bastante adequada para atacar o problema. Alguns fatos justificam esta afirmação: O primeiro, é naturalidade em associar o conceito de casos com o conceito de fragmentos musicais e de atributos musicais de contexto com descritores, a adaptação do fragmento ao contexto com a fase RETAIN do ciclo RBC e assim por diante.

### **3.3.2.2.Reuso de Fragmentos no ImPact**

É no raciocinador em que efetivamente as notas são criadas. Ele percebe as informações disponibilizadas pelo agente ouvinte e fornece as notas criadas para o executor. Esse agente consegue fazer sua tarefa graças a uma forma peculiar de resolução de problemas (Russell 1995). Há um refinamento sucessivo dos objetivos, inicialmente vagos, até chegar à escolha do fragmento a ser efetivamente usado. Dois conceitos são bastante importantes para a realização das

atividades de planejamento: as ações potenciais (*potencial actions* ou *PACTs*) e a memória musical.

A memória musical é um repositório de fragmentos musicais adquiridos por meio da audição de execuções tocadas por músicos importantes. É nela onde há a associação entre as propriedades musicais e de contexto com os fragmentos armazenados. A cada passo durante o processo de criação da linha de contrabaixo, há uma busca na memória musical pelo fragmento que melhor se adequa ao contexto em que ele vai ser inserido, e que possui as propriedades musicais desejadas. Uma vez encontrado, o fragmento é adaptado, caso haja necessidade, para uma ainda melhor adequação ao contexto.

O processo descrito é implementado através de Raciocínio Baseado em Casos (RBC) discutido em detalhes na seção 3.3.2.1. Está é uma forma bastante natural de resolver o problema, visto que os padrões podem ser mapeados diretamente em casos e as propriedades musicais e de contexto, para os índices do caso.

O ImPact implementa um esquema de memória plana para organizar os fragmentos na base de casos. Ele utiliza uma grande quantidade índices para recuperação dos casos: propriedades musicais gerais como, intensidade, contorno de notas, dissonância, densidade, escala e síncope, propriedades musicais intrínsecas ao instrumento, como, estilo do fragmento, possui notas repetidas etc e propriedades de contexto, por exemplo, a sequência de acordes e a posição na música. Como técnica de recuperação, foi utilizado o *k-nn* (cf. seção 3.3.2.1) com  $k=1$ , ou seja, apenas o fragmento mais similar é recuperado. Poucas adaptações estão disponíveis no ImPact. São elas: mudanças de andamento, transposição de notas e mudança de volume das notas. Além dessas, é possível efetuar mudança de altura e de duração das últimas notas do fragmento com o objetivo de suavizar a transição entre os fragmentos.

As *PACTs* (Pachet 1991) são um *framework* de representação de intenções relacionadas ao que o agente pretende tocar imediatamente ou em um momento posterior. Essas intenções podem estar associadas às próprias notas ou a algumas propriedades musicais (síncope, intensidade, escala) das notas. A flexibilidade das *PACTs* favorece a descrição de material musical de acordo com diferentes pontos

de vista e em diferentes níveis de abstração (Ramalho 1994). Por exemplo, os fragmentos que compõem a memória musical representam *PACTs* altamente especializadas (*PACTs* “tocáveis”). Essas *PACTs* contêm descrições de um fragmento melódico em termos de sua sequência de notas (cada uma representada por momento de ataque, duração, altura e volume), além das propriedades musicais e de contexto. Há também aquelas que fornecem apenas alguns indícios sobre o que deve ser tocado (*PACTs* abstratas). Como exemplos delas, pode-se citar: “toque frases sincopadas durante a próxima seção”, “toque cada vez mais alto até o fim da música.”, “tocar no modo dório os próximos dois compassos”. Um outro tipo de *PACTs* (*PACTs* de transformação) descreve as transformações a serem aplicadas a uma *PACT* tocável, como, por exemplo, “aplique uma transposição de semitom acima nessa parte do fragmento”.

De acordo com essa perspectiva, o processo de criação do acompanhamento do contrabaixo foi implementado como uma sucessão de quatro subprocessos de raciocínio que são repetidos até o fim da grade de acordes e são descritos a seguir:

**Segmentação da grade:** O agente estabelece um segmento com respeito a que notas serão computadas e tocadas. Os segmentos possuem como granularidade sequências típicas de acordes. Isso implica a necessidade de armazenar fragmentos de tamanho variável na memória musical.

**Ativação das *PACTs*:** algumas ações potenciais são ativadas de acordo com as informações percebidas do ambiente, isto é, a grade de acordes, o cenário, os eventos ocorridos até o momento, ao que o contrabaixista já tocou numa janela de tempo de um segmento (o último tocado). Essas *PACTs* servem para determinar um conjunto inicial de propriedades musicais (como dissonância, escala, contorno melódico, etc) que irão influenciar as escolhas do fragmento a ser tocado em cada momento.

**Seleção e combinação das *PACTs*:** esse estágio tem o objetivo de combinar propriedades musicais associadas aos *PACTs* ativados, bem como de resolver conflitos entre eles, a fim de chegar a uma *PACT* única. Por exemplo, a *PACT* “use a escala cromática” combinada com “toque notas na direção ascendente” resulta em “toque a escala cromática na direção ascendente”. As *PACTs* ativas



consideradas nesse estágio são aqueles cujos tempos de vida intersectam a sequência de acordes atual.

**Recuperação e adaptação de fragmentos:** A consulta à memória musical é formulada usando a informação da *PACT* resultante do subprocesso anterior mais a descrição do contexto atual. (i.e., o segmento de grade atual, o segmento de grade anterior e os eventos recentes do cenário). Uma vez recuperado, o fragmento é adaptado ao contexto musical corrente utilizando as regras de adaptação disponíveis (transposição, mudança de andamento etc.).

O ImPact contém 256 fragmentos em sua memória musical, além de dezenas de *PACTs* abstratas e de transformação. Foi considerado, em questionário respondido por músicos profissionais, o contrabaixista de Jazz mais completo em comparação com outros sistemas existentes na época, considerando os seguintes aspectos: desenvolvimento, diversificação, fluidez, riqueza melódica, riqueza rítmica, senso harmônico, resultado sonoro e *swing* (Ramalho 1997).

### 3.4.NeurSwing

O NeurSwing (Baggi 1992) é um programa com características bastante particulares. Ele possibilita a investigação de como emerge o *swing* no Jazz, característica indispensável para o estilo, a partir do controle de três parâmetros musicais abstratos: “temperatura”, dissonância e liberdade criativa em relação às indicações da grade de acordes.

Cada um desses parâmetros, pode receber um valor qualquer no intervalo [0,1]. Dessa forma, o valor um para o parâmetro temperatura representa, utilizando o jargão jazzístico, a característica *hot* (instrumentos tocam linhas mais densas, uso mais freqüente de quintas no piano, aumento de andamento etc.), enquanto o valor zero representa *cool* (andamento mais lento, bateria sincopada etc.). Em relação à dissonância, o valor um indica alto grau de dissonância, enquanto zero indica o mais alto grau consonância. Para o parâmetro de liberdade, um indica total liberdade criativa, enquanto zero indica conformidade total com a grade de acordes.

A grade de acordes de uma determinada música serve de entrada para o sistema. A partir dela, o sistema constrói uma rede neural artificial (Mitchell 1997; Duda 2000), chamada de rede harmônica, representando as informações musicais contidas na grade e estabelecendo uma segmentação por compassos. Os parâmetros musicais (temperatura, dissonância e liberdade) servem de entrada para outra rede neural artificial (chamada de estilística, uma vez que envolve questões relativas ao estilo). Cada saída da rede irá atuar na rede harmônica, ponderando o valor estabelecido para cada um dos três parâmetros (temperatura, dissonância e liberdade) para cada fragmento musical existente. A escolha é feita utilizando uma função aleatória ponderada pelo valor dos parâmetros para cada fragmento. A função também utiliza a informação do próximo acorde a ser utilizado para impedir que o mesmo padrão seja estabelecido consecutivamente.

O NeurSwing utiliza 16 fragmentos distintos de piano e 22 fragmentos de bateria (o baixo não é produzido através do encadeamento de fragmentos) podendo possuir tamanhos distintos. Os critérios de adaptação utilizados são bastante poderosos, são eles: transposição, inserções de notas de passagem, inversão de acordes e mudança de notas em acordes a fim de causar efeitos musicais específicos como por exemplo, trocar a quinta do acorde por uma quinta bemol com o objetivo de aumentar a tensão melódica.

### **3.5.Considerações Finais**

Neste capítulo, foram apresentados alguns dos sistemas existentes para a geração automática de linhas musicais. A Tabela 1 resume as informações mais relevantes sobre cada um deles.

Sistema	Instrumento	Estilo	Técnica principal	Granularidade	Janela	Ambiente	Tamanho(frag.)	Descrição(frag.)	Recuperação	Adaptação
<b>Johnson-Laird</b>	Contrabaixo	<i>Standard Jazz</i>	Gramática Regular	Tempo	Uma nota	Grade de acordes	-	-	-	-
<b>Cybernetic Composer</b>	piano,baixo bateria	Rock, <i>Jazz</i> e Ragtime	APD	Compasso	Não possui	Grade de acordes	-	-	-	-
<b>Giomi</b>	Saxofone	<i>Standard Jazz</i>	APD	Compasso	Não possui	Grade de acordes	-	-	-	-
<b>Spector</b>	Saxofone	Bebop	Algoritmo Genético	4 comps ( <i>trading fours</i> )	Não possui	Compassos tocados pelo usuário	-	-	-	-
<i>GenJam</i>	Saxofone	Jazz	Algoritmo Genético	4 comps ( <i>trading fours</i> )	Não possui	Compassos tocados pelo usuário	-	-	-	-
<b>Band-in-a-box</b>	Piano, contrabaixo, guitarra, bateria e cordas	Diversos estilos	Reuso de Fragmentos	Compasso	Não possui	Grade de acordes	Variável (de um a oito tempos)	Poucas propriedades de contexto	Seleção aleatória ponderada	Simples (transposição, mudança de andamento etc)
<b>NeurSwing</b>	Piano, bateria e contrabaixo	<i>Standard Jazz</i>	Reuso de Fragmentos (Redes Neurais)	Compasso	compasso adiante	Rede harmônica e rede estilística	Variável	Dissonância, temperatura e conformidade	Seleção aleatória ponderada pela rede estilística	Complexa (inserção de notas, manipulação de tensão etc)
<b>ImPact</b>	Contrabaixo	<i>Standard Jazz</i>	Reuso de Fragmentos (Regras + RBC)	<i>Chord chunks</i>	<i>Chord chunk</i> anterior	Grade de acordes + cenários	Variável	Rica (propriedades musicais e de contexto)	RBC (similaridade usando k-nn)	Simples (transposição andamento etc)

Tabela 1 – Resumo dos sistemas citados no texto

## 4. O Modelo Proposto

Neste capítulo, descreveremos a abordagem escolhida para enfrentar o problema da geração de acompanhamento rítmico para o violão, descrevendo as escolhas de projeto feitas e os motivos de tais escolhas.

### 4.1. Escolhendo a Metodologia a ser Adotada

A fim de abordar o problema da geração automática de acompanhamento rítmico para o violão, adotaremos a metodologia de reuso de fragmentos musicais (cf. seção 3.3). Dois argumentos principais justificam essa escolha. O primeiro reside no fato de que tal abordagem é muito mais extensível e modular que a sua concorrente (nota por nota). Por exemplo, no caso da geração a partir de gramáticas ou regras, ao contrário do que acontece nas abordagens que reutilizam fragmentos, para cada estilo e instrumento, além das próprias notas, detalhes como, tessitura dos instrumentos, restrições por questões anatômicas também tem que ser levadas em conta. Na realidade, fazer qualquer modificação de requisitos nesse tipo de sistemas provoca a necessidade de refazer todo o trabalho de aquisição e formalização de conhecimento. Abordagem baseada fragmentos simplifica o processo de aquisição de conhecimento (cf. seção 3.3), uma vez que muito do conhecimento já está incorporado nos próprios fragmentos.

O segundo e principal argumento, é a naturalidade com que essa abordagem lida com a geração de ritmos. É através da noção de recorrência que percebemos o ritmo (Keifer 1984), e, por meio de *padrões rítmicos* (no real sentido da palavra: fragmentos recorrentes), popularmente chamados de “batidas”, é que freqüentemente ocorre o aprendizado do acompanhamento rítmico no violão popular (Sandroni 1985).

De forma mais específica, adotaremos a abordagem proposta no sistema ImPact (cf. seção 3.3.2), em detrimento das outras abordagens. Uma vasta documentação

disponível, o acesso e a disponibilidade do próprio autor, professor Geber Ramalho, para esclarecer eventuais dúvidas e a simplicidade de mapeamento de padrões rítmicos em casos são algumas dos argumentos que justificam essa escolha. Além disso, os outros sistemas citados (*Band-in-a-box* e *NeurSwing*) baseiam-se, mesmo que de forma reduzida, em decisões aleatórias, que não correspondem a bons exemplos para os usuários do D'Accord Violão que desejam aprender a como tocar um acompanhamento rítmico.

É importante deixar claro, que com essa afirmação, não descartamos a hipótese de que o ser humano utiliza componentes aleatórios no momento da decisão de como tocar um certo acompanhamento rítmico. Estamos, sim, afirmando que, baseados no fato de não termos o conhecimento de como ocorre efetivamente esse processo de decisão no cérebro humano, uma hipótese baseada em determinismo é de maior interesse didático.

Porém, a utilização da abordagem ImPact, evidentemente, não pode ser feita sem adequações. Em decorrência das mudanças na escolha do instrumento (de baixo para violão), do estilo musical (de Jazz para Bossa Nova – escolha discutida na próxima seção (4.2)), de informações de ambiente (banda para voz-e-violão) e sobretudo, na natureza da linha musical (melodia para ritmo) se fazem necessários:

- Uma nova biblioteca de fragmentos musicais;
- novos atributos para descrever esses fragmentos;
- novas informações sobre o contexto musical (características da melodia e da grade de acordes que são relevantes); e
- novas regras para guiar a escolha de que fragmento usar.

Vale salientar a impossibilidade de afirmar qual das duas tarefas (geração de acompanhamento melódico para baixo *versus* geração de acompanhamento rítmico para violão) representa uma dificuldade maior. Uma linha de baixo transita em um ambiente mais complexo, onde muitas variáveis devem ser levadas em consideração (daí a grande quantidade de atributos musicais e de contexto

existentes no ImPact). Em contrapartida a essa dificuldade, há disponível uma vasta literatura em teoria musical que auxilia no processo de construção de uma linha melódica. No caso do acompanhamento rítmico, alguns desses elementos são bastante difíceis de elicitar, quando em comparação com o as linhas de baixo para Jazz. Isso ocorre principalmente pela escassa formalização do conhecimento relativo à geração de acompanhamentos rítmicos (cf. seção 2.7).

#### **4.2. Escolhendo o Estilo Musical: Bossa Nova**

A necessidade de escolher um estilo musical neste momento se deve ao fato que necessitamos de uma forma empírica de avaliar os resultados obtidos nesse trabalho. O estilo a ser escolhido deve preencher os requisitos de popularidade e difusão, permitindo a análise dos resultados de forma mas fácil (grande quantidade de pessoas capacitadas a avaliar o sistema com propriedade) Inegavelmente, nenhum outro movimento foi tão revolucionário na música brasileira quanto a Bossa Nova. Surgido por volta de 1958, mudou de forma radical a maneira de compor melodia, letra, harmonia, ritmo e arranjos, dando a música brasileira uma notoriedade internacional não alcançada em nenhuma outra época (Américo 2004).

Além da importância para a cultura brasileira, um dos fatores predominantes para a escolha desse estilo foi o papel do acompanhamento rítmico criado por João Gilberto na disseminação daquele, fato citado por diversos músicos; personagens importantes da própria Bossa Nova, como Ronaldo Bôscoli:

*Foi através dele, através do violão dele que nasceu a Bossa Nova. Ele definiu com aquela batida dele o que só havia sido indicado por Johnny Alf, Lúcio Alves e os outros grandes nomes da Pré-Bossa. O disco Chega de Saudade foi definitivo (Garcia 1999).*

Roberto Menescal: “Sem o violão de João não haveria Bossa Nova” (Garcia 1999) e Tom Jobim:

*O grande salto do Samba-Canção para a Bossa Nova foi a batida do João. Essa batida do João hoje em dia está incorporada. O universo todo conhece essa batida (Garcia 1999).*

E é em homenagem a ele, João Gilberto, um dos maiores expoentes da música popular brasileira, que batizamos o sistema com o nome de *Cyber-João*.

As próximas seções do capítulo são destinadas à discussão sobre as principais adaptações necessárias à reutilização do modelo ImPact no contexto em que será utilizada o *Cyber-João*. Como foi dito na seção 4.1, essas adaptações são: a construção de uma nova biblioteca de fragmentos, os atributos para indexação e recuperação dos fragmentos, as novas regras para refinar as escolhas dos fragmentos e os novos dados percebidos através do ambiente.

### 4.3. Escolhendo os Padrões Rítmicos

A tarefa de escolher os padrões rítmicos utilizados na Bossa Nova pode ser realizada de diferentes formas. Pode-se adquiri-los diretamente, escutando um *corpus* de músicas e transcrevendo os padrões encontrados “de ouvido”. Essa tarefa é extremamente desgastante, uma vez que o *corpus* citado deve possuir o maior tamanho possível (além da necessidade de que esse *corpus* seja representativo). Uma outra forma de adquirir os padrões é utilizar aprendizado de máquina para extraí-los automaticamente (Cope 1991; Rolland 1996). A dificuldade, nesse caso, consiste em implementar um sistema que faz a tarefa correta e eficientemente.

Felizmente, a existência de dois trabalhos musicológicos, um feito por Carlos Sandroni (Sandroni 1985) e o outro, por Walter Garcia (Garcia 1999) simplificaram sobremaneira a tarefa de aquisição desses padrões. Em ambos, é feita uma análise da obra de João Gilberto sob o ângulo dos padrões rítmicos criados por ele. A partir desses estudos, é possível perceber algumas características rítmicas bastante peculiares.

Apenas dois tipos de eventos são encontrados nos padrões: um evento executado pelo dedo polegar, comumente chamado de bordão, que ataca sempre a nota mais grave do acorde; e o evento executado pelos dedos indicador, médio e anular juntos, atacando três cordas em posições consecutivas (contado as cordas de baixo para cima, primeira, segunda e terceira, ou segunda, terceira e quarta, a depender da formação do acorde), evento conhecido popularmente como “puxada”.

Os padrões são sempre representados por compassos binários (assinatura 2/4), a exemplo da representação utilizada classicamente no samba.

A Figura 14 exibe o padrão rítmico clássico criado por João Gilberto. As notas na clave de sol indicam os eventos de “puxada”, enquanto as notas na clave de fá indicam os eventos de bordão. A partir desse padrão principal, o próprio João Gilberto cria uma grande quantidade de variantes.

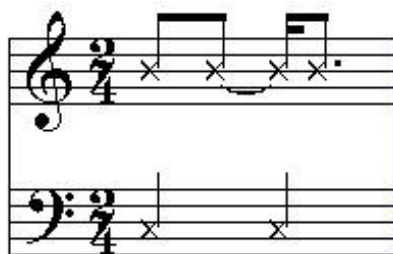


Figura 14 – Principal padrão rítmico criado por João Gilberto. A clave de sol indica os ataques das puxadas e a clave de fá indica os ataques do bordão.

Em (Sandroni 1985) obtemos a explicação para essa grande variedade de padrões:

*Note-se primeiramente que ele [o padrão] apresenta uma espécie de vazio no seu último meio tempo. Ao contrário, nos três quartos iniciais do compasso, não se passam dois quartos de tempo sem que uma figura rítmica seja articulada (Sandroni 1985).*

A Tabela 2 confirma a afirmação. De fato, os “quartos de tempo” 7 e 8 não possuem eventos (articulações) associados.

Quarto de tempo	1	2	3	4	5	6	7	8
<b>Puxada</b>	X		X			X		
<b>Baixo</b>	X				X			

Tabela 2 – Eventos executados por quarto de tempo

O texto continua:

*De que tal coisa tende a ser sentida como um vazio pelo executante é testemunha o fato de que os violonistas iniciantes tendem regularmente a “preenchê-lo” com um terceiro toque do polegar, no oitavo quarto de tempo...(Sandroni 1985).*

Este fato fica claro na Figura 15:



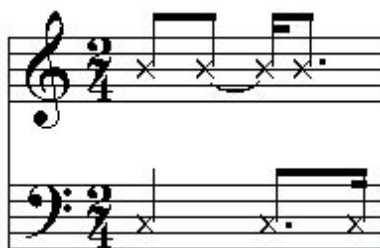


Figura 15 – Erro comumente praticado por iniciantes: executar um evento de bordão no ultimo quarto de tempo.

E Sandroni finaliza:

*Mas a solução adotada por João Gilberto e consagrada pela boa prática foi, em vez disso, antecipar a primeira “puxada” do tempo seguinte (justamente a que acontecia simultaneamente ao baixo e, portanto, do ponto de vista da articulação, nula) (Sandroni 1985).*

Encadeando o fragmento rítmico com ele mesmo, temos como resultado o padrão rítmico mostrado na Figura 16, de onde surge o padrão cíclico da Figura 17 (as duas ligaduras soltas no início e no fim do padrão representam o fato de o padrão ser cíclico), cujo fim pode ser ligado ao seu próprio início.

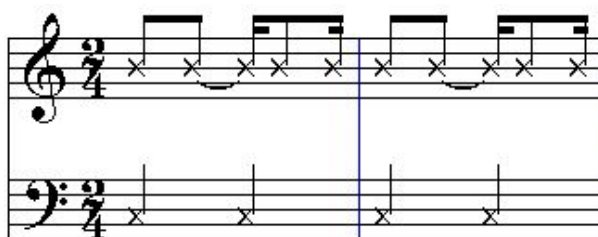


Figura 16 – O padrão rítmico justaposto a ele mesmo

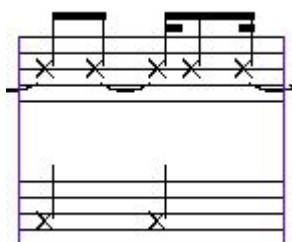


Figura 17 – Padrão cíclico resultante do padrão clássico

A justaposição do padrão clássico (Figura 14) e o padrão cíclico (Figura 17), abrem espaço para uma série de variações cujo desenho completo perfaz sempre dois compassos, onde o segundo é sempre igual ao mesmo padrão clássico

alterado apenas pela última nota invariavelmente sincopada<sup>4</sup> (corresponde ao padrão cíclico sem a ligadura de início). O resultado é um conjunto de seis novos padrões rítmicos (ver Anexo A para uma lista completa), um dentre eles exemplificado na Figura 18:

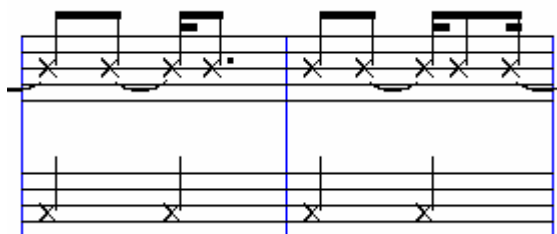


Figura 18 – Exemplo de padrão resultante da combinação dos padrões das Figura 14 e Figura 17

Vale a pena salientar que, embora os dois primeiros padrões citados (o clássico e cíclico) sejam compostos por apenas um compasso, na prática eles sempre são tocados em blocos de tamanho par.

Sandroni cita, ainda, que há uma clara diferença nas performances de Bossas Novas lentas (menos que 80 semínimas por minuto) e rápidas. Grande parte dos padrões só pode ser utilizada em músicas rápidas.

Três casos especiais não são contemplados pelos padrões citados até o momento, são eles: o início da música, um momento de ritmo harmônico muito alto e os momentos de “virada”.

Para o início da música, existem duas alternativas. Ou ela acontece anacrusticamente<sup>5</sup>, tocando uma semicolcheia isolada antes do início do primeiro compasso (a semicolcheia estaria ligada ao padrão rítmico desejado), ou, prática

<sup>4</sup> Entende-se por síncope a nota tocada que é produzida em um tempo fraco, prolongado-se ao tempo forte seguinte.

<sup>5</sup> Em referência à Anacruse, notas que, no início de peça musical, antecedem o primeiro tempo forte do compasso inicial.

mais comum, começa-se teticamente<sup>6</sup>, esquecendo a ligadura e tocando o compasso inicial de modo ligeiramente diferente de todo o resto da música. Essa tendência a acentuar o tempo forte no início da música também se verifica em acompanhamentos de outros estilos da música popular brasileira e, portanto, devido à sua popularidade, adotamos essa alternativa nesse trabalho.

Ritmo Harmônico é um fator que indica a velocidade com que ocorrem mudanças de acorde em um determinado segmento musical. No caso da Bossa Nova, e da grande maioria da música popular brasileira, a mudança de acordes só pode ocorrer na cabeça de cada tempo do compasso. Quando a harmonia muda duas vezes por compasso, fato não muito comum no estilo, costuma-se utilizar dois padrões especiais, mostrados na Figura 19 e na Figura 20.

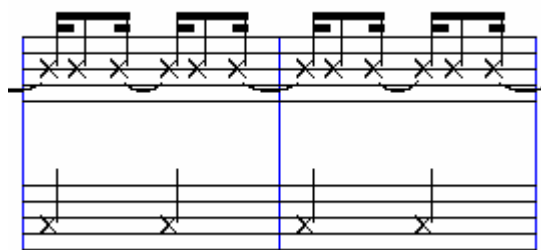


Figura 19 – Padrão rítmico especial. Utilizado em momentos de ritmo harmônico alto

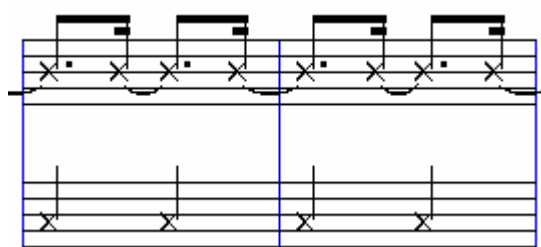


Figura 20 – Padrão rítmico especial. Utilizado em momentos de ritmo harmônico alto

No caso das “viradas”, nada é citado na literatura. Tivemos, então, que utilizar o processo de aquisição de padrões através de escuta e transcrição (c.f. seção 4.2).

Analisando um *corpus* composto por três discos com gravações de João Gilberto (Gilberto 1995; Gilberto 1998; Gilberto 2000), pudemos perceber que os padrões

---

<sup>6</sup> Que tem início no começo do compasso (tético).

de viradas são compostos a partir da justaposição do primeiro compasso de um padrão rítmico convencional, com o segundo compasso do padrão especial mostrado na Figura 19, formando padrões como o apresentado da Figura 21. Porém, ao contrário do que se possa pensar, nem todos os padrões convencionais possuem uma virada correspondente, apenas aqueles que não possuem eventos de puxada no último quarto de tempo. Os padrões de virada também são cíclicos.

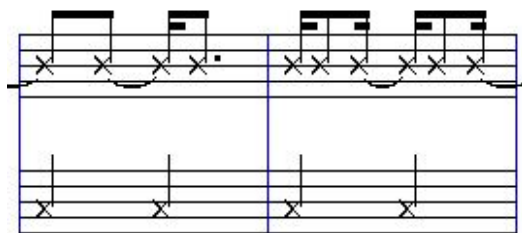


Figura 21 – Exemplo de padrão de virada

#### 4.4.Representação e Natureza dos Padrões

Baseado nas considerações supracitadas e utilizando os critérios estabelecidos para análise para dos Sistemas de Acompanhamento Automático, optamos por utilizar fragmentos de tamanho fixo, com granularidade de dois compassos binários, já que grande parte dos padrões é representada em dois compassos e, aqueles que não são, geralmente são usados em blocos com esse tamanho.

Não implementaremos métodos de adaptação de casos. Qualquer adaptação feita mudaria a estrutura do padrão, o que não é recomendável. Os padrões, portanto, devem ser reusados exatamente como tocados por João Gilberto.

Utilizaremos uma janela de tempo de tamanho de dois compassos binários. Ela terá sempre a informação do último padrão escolhido. Com essa janela, é possível manter a coerência das escolhas atuais com o que já foi tocado anteriormente.

##### 4.4.1.Atributos Associados aos Padrões

Embora não haja registro na literatura associando objetivamente aplicação de padrões a propriedades musicais e contexto onde são utilizados, é possível, a partir do que já foi discutido até o momento, perceber certas “sugestões” sobre o

tema. A aquisição dessas informações foi feita através de estudos da teoria musical e em conversas freqüentes com especialistas. Em ambos os casos, percebemos que há um número restrito de atributos associados à escolha de padrões, fato intrinsecamente relacionado à falta de formalização do conhecimento (c.f. seção 2.5.1).

O nosso modelo foi proposto com uma pequena quantidade de atributos. Ele possui dois atributos de contexto e quatro atributos envolvendo propriedades musicais. Os atributos de contexto são *ritmo harmônico* e *andamento*, explicados a seguir.

O *ritmo harmônico*, como já dito, indica a velocidade com que ocorrem mudanças de acorde em um determinado segmento musical. Usando uma segmentação fixa de dois compassos, como as mudanças de acordes só ocorrem nas cabeças dos tempos, percebemos que existem 15 valores diferentes possíveis para o ritmo harmônico, correspondentes à existência ou não de uma mudança de acorde em cada um dos quatro tempos do segmento. Por exemplo, em um segmento onde ocorre mudança de acordes nos tempos 1 e 3, associa-se um ritmo harmônico de valor 5 ( $1*1+2*0+4*1+8*0$ ). No caso de existirem novos acordes nos tempos 2, 3, 4, o valor do ritmo harmônico será 14 ( $1*0+2*1+4*1+8*1$ ) e assim por diante. A Tabela 3 contém a lista exaustiva das possibilidades de ritmo harmônico. É importante notar que esse é um atributo qualitativo, e não quantitativo.

<b>Ritmo</b> <b>Harmônico</b>	<b>1°. Compasso</b>		<b>2°. Compasso</b>	
	<b>1°. Tempo</b>	<b>2°. Tempo</b>	<b>1°. Tempo</b>	<b>2°. Tempo</b>
1	X			
2		X		
3	X	X		
4			X	
5	X		X	
6		X	X	
7	X	X	X	
8				X
9	X			X
10		X		X
11	X	X		X
12			X	X
13	X		X	X
14		X	X	X
15	X	X	X	X

Tabela 3 – Todos os valores possíveis de Ritmo Harmônico associados à presença de acordes nos tempos dos compassos.

O *andamento*, que indica a velocidade de execução da música. Pode assumir apenas dois valores: “Lento”, quando o valor for menor que 80 semínimas por minuto e “Rápido”, caso contrário.

As propriedades musicais consideradas são as seguintes: *frequência de uso*, *densidade*, *início* e *virada*, explicadas a seguir.

A *frequência de uso* indica o quão frequentemente esse padrão é usado na Bossa Nova. Pode assumir valores em um intervalo de 1 a 5, indicando, respectivamente, mínima e máxima importância.

A *densidade*, indica o número de eventos existentes no padrão. Como, a quantidade de bordões é igual para todos os padrões, a densidade conta a

quantidade de puxadas no padrão. Os valores são divididos em três grupos: “baixa”, “média” e “alta”.

O *início* indica se o padrão foi ou não tocado no início da música. Apenas padrões não cíclicos o podem ser utilizados nesse contexto. E por fim, *virada*, indica se um determinado padrão é ou não um padrão de virada.

Cada um dos atributos deve possuir um peso associado. Esse peso indicará a importância do atributo para a do padrão. Há duas formas principais de associar o peso a um atributo (Kolodner 1993): de forma automática, a partir de uma grande quantidade de exemplos, ou de forma manual, a partir de conversas com especialistas. Optamos pela última, principalmente por não dispormos de exemplos para alimentar a escolha automática. Utilizamos o intervalo de um (mínima importância) a cinco (máxima importância) para cada atributo, como mostra a Tabela 4.

Atributo	Peso
Ritmo harmônico	4
Andamento	5
Frequência de uso	2
Densidade	1
Início	5
Virada	3

Tabela 4 – Pesos associados ao atributos

A Tabela 5 ilustra um dos padrões rítmicos utilizados com seus respectivos valores de atributos.

Padrão Rítmico			
			
Contexto Musical		Propriedades Musicais	
Atributo	Valor	Atributo	Valor
Ritmo Harmônico	6 (2º e 3º tempos)	Densidade	Média
		Frequência de uso	4
Andamento	Rápido	Virada	Não
		Início	Não

Tabela 5 – Exemplo de atributos associados a um padrão rítmico.

#### 4.4.2. Técnica de Recuperação de Padrões Utilizada

A técnica de recuperação de padrões que adotada consiste em um uso combinado de raciocínio baseado em regras e raciocínio baseado em casos, como no sistema ImPact (cf. seção 3.3.2).

O RBC constitui a memória musical do sistema. Existem, até o momento, 21 padrões rítmicos catalogados, entre os padrões cíclicos, especiais, de início de música e de virada, estruturados em uma memória plana (cf. seção 3.3.2.1). Essa estrutura apresenta grande acessibilidade e, com essa pequena quantidade de padrões, não provoca gargalo de eficiência. Para recuperar os casos, utilizamos o método *k-nearest neighbors* (seção 3.3.2.1) com  $k=1$ .

O raciocínio baseado em regras determina quais as características musicais que um padrão deve possuir para ser inserido em um determinado segmento. O processo de ativação das regras foi implementado de um modo bastante simples. Para cada segmento, todas as regras são disparadas. Cada regra está associada a



uma ou mais propriedades musicais dos padrões. Quando uma regra é disparada, há uma modificação no valor dos atributos musicais que serão usados para recuperar um determinado caso. O fato de não existirem regras conflitantes permite essa implementação trivial.

Até o momento, foram criadas seis regras que abordam situações como início de música, viradas e algumas outras situações especiais. No Quadro 1 vemos dois exemplos de regras usadas (ver Anexo B para uma lista completa).

<b>Regra: Virada</b> <b>SE</b> (melodia no último tempo do segundo compasso $\leq 2$ notas) <b>E</b> (se há um esquema V7-I entre o último acorde do segmento e o primeiro do próximo) <b>ENTÃO</b> virada = sim; “use as outras propriedades musicais do padrão anterior”
<b>Regra: Manipula início</b> <b>SE</b> (é o início da música) <b>ENTÃO</b> início = sim; densidade = baixa

Quadro 1 – Dois exemplo de regras implementadas

Nas Figura 22 e Figura 23 é mostrado um exemplo de como acontece a aplicação da regra de virada em um trecho da música *Desafinado*. A Figura 22 exhibe a avaliação das condições do ambiente.

The image displays a musical score for the song "Desafinado" in 2/4 time. The melody is written on a treble clef staff, and the bass line is on a bass clef staff. Chords are indicated above the staff: A7M, B<sup>b</sup>dim, Bm7, E7(13), A7M, and Am7. A blue box labeled "Segmento" highlights a portion of the melody starting from the Bm7 chord. Above the "Segmento" box, the chord progression is labeled: IIm7, V7, and I. A red oval highlights the V7 chord (E7(13)). Below the "Segmento" box, a label "Número de notas no tempo" points to the number "0", indicating the number of notes in the final measure of the segment.

Figura 22 – Ilustração do emprego da regra virada (avaliação das condições).

Nota-se que se configuram as condições para virada (pequena quantidade de notas no último tempo do último compasso do segmento e esquema de acordes da forma V7-I entre o último acorde do segmento atual e o primeiro do seguinte).

A Figura 23 mostra a escolha efetiva a partir daquela avaliação. Percebe-se que o primeiro compasso da virada (representada pelos dois últimos compassos da pauta inferior) é idêntico ao primeiro compasso do segmento anterior. Isso se deve ao conseqüente da regra (“use as outras propriedades musicais do padrão anterior”).

Figura 23 – Ilustração do emprego da regra virada (escolha do padrão)

#### 4.5.Considerações Finais

Neste capítulo, mostramos as principais escolhas tomadas para enfrentar o problema de acompanhamento rítmico. O modelo explicado é uma adaptação do modelo usado no sistema ImPact, com as devidas modificações para uma melhor adequação ao novos estilo, instrumento e dimensão musical. No capítulo precedente explicaremos em detalhes as decisões de implementação e os resultados obtidos.

## 5. Cyber-João: Implementação e Resultados

Neste capítulo, descreveremos os principais aspectos envolvendo a implementação do sistema *Cyber-João*. Comentaremos o protocolo de análise de resultados e discutiremos os resultados obtidos.

### 5.1. Visão Geral

A Figura 24 apresenta um diagrama esquemático da arquitetura do sistema implementado.

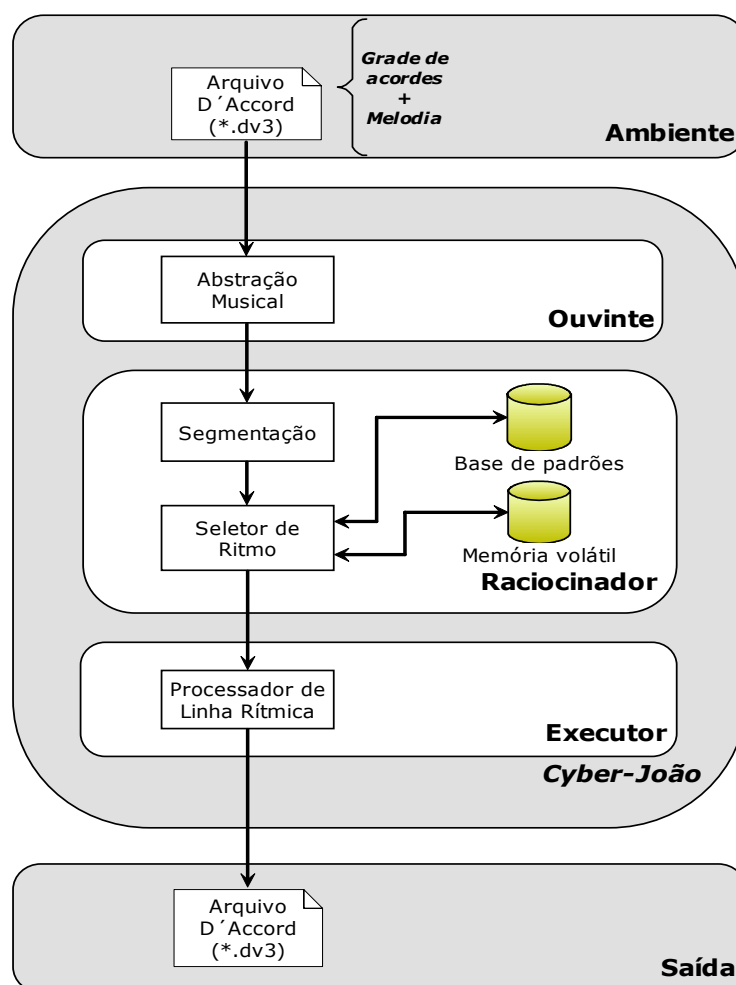


Figura 24 – Diagrama esquemático arquitetural do nosso modelo

O sistema implementado é dividido em três grandes blocos: a entrada, ou ambiente, fonte das informações da música para a qual se deseja criar o acompanhamento rítmico; o *Cyber-João*, módulo principal do sistema, onde ocorre todo o processamento e a saída para onde vão os resultados do processamento.

De maneira similar ao ImPact, dividimos a parte principal do sistema (o bloco como nome de *Cyber-João*) em três partes, de acordo com a correspondência de sua atividade com àquelas encontradas na modelagem de Agentes Inteligentes (Russell 1995). O *Ouvinte* interpreta as informações do ambiente, o *Raciocinador* escolhe que padrões rítmicos devem aparecer em cada segmento e o *Executor* faz junção entre os resultados gerados pelo *Raciocinador* e as demais informações que compõem a música, preocupando-se, também, com a expressividade da performance (dinâmica, articulação, rubato etc).

De forma geral, o sistema funciona da seguinte forma. Inicialmente, ele recebe como entrada um arquivo no formato proprietário D'Accord Violão sem um acompanhamento rítmico associado. As informações relevantes contidas no arquivo dv3 (a grade de acordes e a melodia) são então utilizadas no processo chamado *abstração musical* para criar uma estrutura de dados que reifica uma representação da música, onde as notas da melodia e os acordes da grade são hierarquizados em estruturas maiores, os tempos, e estes, em compassos. Isso é necessário porque os eventos de nota e acordes no formato do D'Accord Violão são representados apenas pelos momentos de acionamento e desligamento das notas, da mesma forma que o formato MIDI. Adicionalmente, essa estrutura de dados (que também podemos chamar de abstração musical) contém os atributos de contexto que também são extraídos do arquivo dv3 e necessários ao processo de geração automática, citado no capítulo anterior.

Além dos atributos ritmo harmônico e andamento, são também extraídos outros atributos que ajudarão a compor as informações necessárias à aplicação das regras de refinamento. Os seguintes atributos são extraídos:

- A *densidade melódica*, que corresponde ao número de notas que compõe a melodia, para cada tempo;

- a *categoria dos acordes*, que corresponde à informação de qual a categoria do acorde ( maior, menor, sétima, diminuto etc.) (Chediak 1986) e
- o *grau do acorde*, que corresponde ao intervalo entre a tônica do acorde e a tonalidade da música (Chediak 1986). Alguns valores possíveis são: I, II, III, SubV7, V7 etc.

A abstração musical é então segmentada (na fase de *segmentação*) de acordo com as necessidades do sistema. Como foi dito anteriormente, a música é dividida em segmentos de dois compassos binários. No caso em que a abstração musical possua quantidade ímpar de compassos será acrescentado um compasso vazio ao final da abstração.

O resultado do processo de *segmentação* (uma seqüência de segmentos de dois compassos binários) alimenta o *seletor de ritmo*, unidade que efetivamente escolhe que padrão rítmico deve ser selecionado para cada segmento. Ele faz uso de duas memórias, a *memória musical* (discutida em detalhes na seção 3.3.2), onde estão armazenados os padrões rítmicos e seus respectivos atributos, e a *memória volátil*, onde são armazenadas as informações relativas à janela de tempo, constituída pelas informações do contexto do segmento anterior e o padrão rítmico escolhido para aquele segmento.

O resultado obtido pelo *seletor de ritmo* (uma seqüência de padrões rítmicos associados a cada segmento) é utilizado no *processador de linha rítmica* onde são adicionadas informações relativas à expressividade.

Em relação à expressividade, devido ao fato de estar fora do escopo dessa dissertação, implementamos apenas pequenas variações na duração dos acordes de acordo com o momento de ataque de cada evento do mesmo (da mesma forma como é feito no programa *Encore*<sup>7</sup>). Por fim, um novo arquivo dv3 é gerado consistindo no arquivo anterior adicionado de uma linha rítmica.

---

<sup>7</sup> Encore (versão 4.5) é marcar registrada da *GVOX Inc.* (<http://www.gvox.com>). Todas as informações foram retiradas do manual do usuário do programa.

Por fim, as informações são gravadas em um outro no formato D'Accord Violão, este contendo adicionalmente as informações da linha rítmica gerada pelo *Cyber-João*.

A implementação da maioria dos processos vistos acima é bastante simples<sup>8</sup>. De fato, a quase totalidade da complexidade do sistema está no processo *seletor de ritmo*, o coração do sistema que no caso do *Cyber-João* possui dois módulos, um simples *Disparador de Regras* e uma instanciação de um *Motor de Raciocínio Baseado em Casos*.

A tarefa do *Disparador de Regras* é simplesmente executar a ação das regras (Anexo B) cujas condições foram aceitas. Na prática, esse módulo é executado antes do *Motor de Raciocínio Baseado em Casos* (cf. seção 5.2) ajudando a refinar a busca na base de casos.

## 5.2. Seletor de ritmo: Motor de Raciocínio Baseado em Casos

O motor é composto por uma classe principal (*CBREngine*), que possui todos os métodos necessários para o uso de uma aplicação RBC, como, por exemplo, recuperar casos a partir de seus atributos, adaptar e armazenar os casos etc. Para criar uma nova aplicação, é necessária apenas a implementação de alguns métodos nas classes abstratas levando em consideração características da aplicação desejada. A Figura 25 ilustra o diagrama de classes UML (Booch 1999) do motor.

---

<sup>8</sup> Faz-se uma ressalva aqui ao processo segmentação que, embora extremamente simples no caso específico do Sistema *Cyber-João* pode ter enorme complexidade em aplicações que segmentação mais sofisticada.

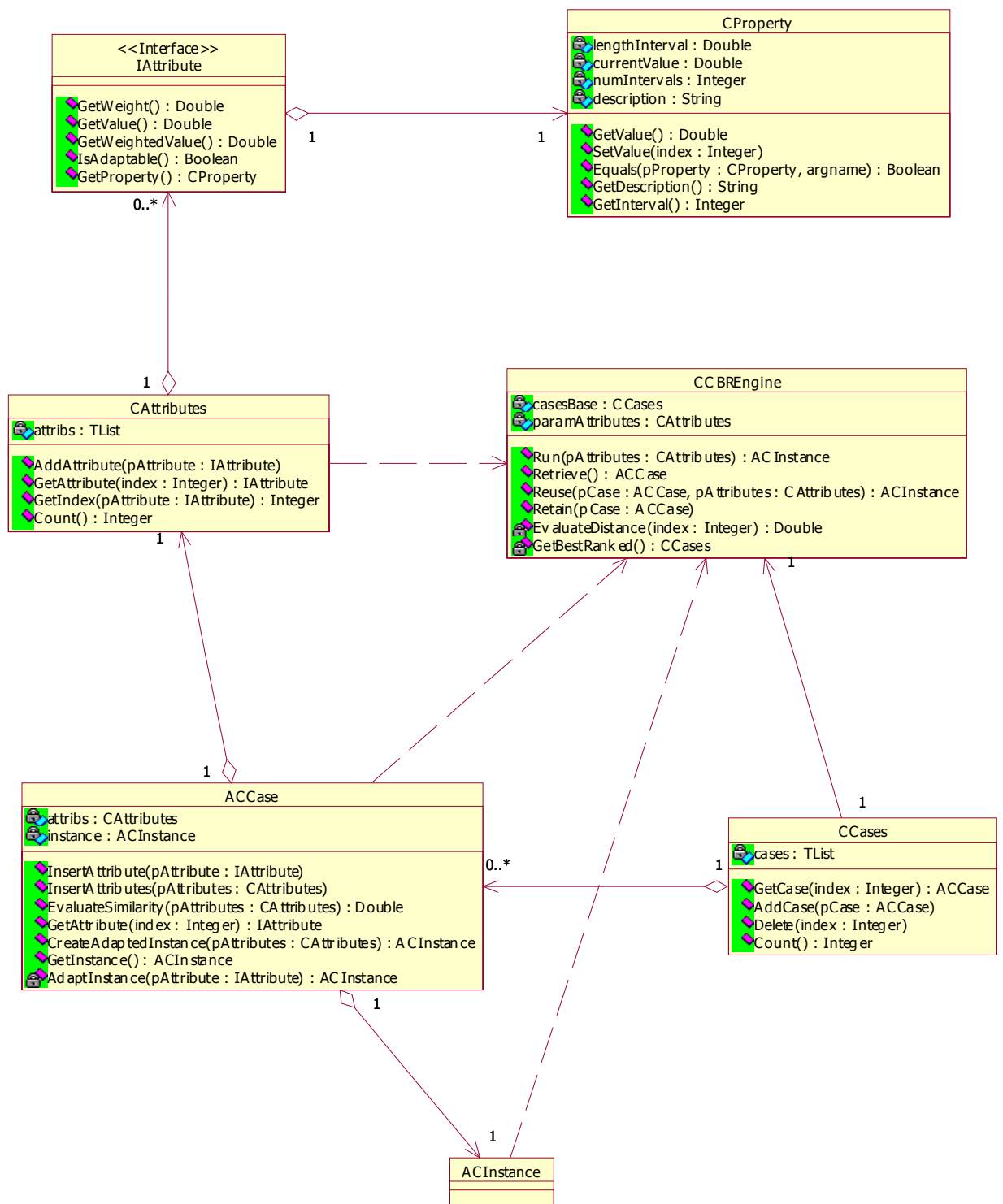


Figura 25 – Diagrama de classes do motor RBC (UML)

Uma breve descrição da suas classes básicas e seu funcionamento aparece a seguir:

- *CProperty* representa um descritor, isto é, um atributo com seu valor associado, definido para indexação dos casos, por exemplo, Ritmo harmônico = 15, Andamento=Rápido. *CProperty* é uma classe concreta.
- *IAttribute* é uma interface para um determinado atributo. Um atributo é um par composto de uma propriedade (*CProperty*) e um peso a ela associado. Por exemplo, (ritmo harmônico = 15, peso = 4).
- *CAttributes* representa um conjunto de atributos, é uma classe concreta. No caso do *Cyber-João* essa classe sempre conterá 6 atributos (Ritmo harmônico, Andamento, Frequência de uso, Densidade, Início e Virada) .
- *ACInstance* é o resultado de uma consulta RBC. É uma classe abstrata sem métodos definidos ou implementados. Serve apenas como pai de qualquer instância concreta. Por exemplo, no nosso caso, a classe *CPadraoRitmico*, que possui as informações de evento dos padrões rítmicos, bem como a capacidade de colocá-los no formato proprietário D'Accord Violão, implementa *ACInstance*
- *ACCase* é uma classe abstrata que representa um caso. É composta por um conjunto de atributos (*CAttributes*) associados a uma instância (*ACInstance*). É nessa classe que é definido o método de similaridade entre os casos. Para o *Cyber-João*, implementamos *k-nn*, com  $k=1$ .
- *CCases* é uma classe concreta, pode ser vista como um conjunto de casos.

Como é possível perceber pela descrição, este motor tem os requisitos de engenharia de software necessários (extensibilidade, modularidade e reusabilidade (Sommerville 2000)) para que possa ser usado em outras classes de problemas e aplicações, bastando para isso implementar apenas algumas poucas classes.

### 5.3. Metodologia de Testes

A fim de verificar o desempenho do sistema *Cyber-João*, fez-se necessária, a concepção de um ambiente de testes. A natureza pouco formal deste trabalho (e dos trabalhos que envolvem criatividade em geral), optamos por um ambiente em



que foi possível realizar uma validação pragmática dos resultados. Este ambiente é composto pelo sistema *Cyber-João* e por mais dois sistemas implementados com diferentes abordagens: o *Crazy-João* e o *João-in-a-Box*.

O primeiro seleciona os padrões a serem usados na música de forma totalmente aleatória, servindo de patamar mínimo de qualidade, enquanto o *João-in-a-Box*, utiliza os mesmos critérios de seleção do software *Band-in-a-Box*, ou seja, busca aleatória, restrita por regras e guiada por pesos (c.f. seção 3.3.1).

O grande benefício das abordagens utilizadas nesses dois novos sistemas é o fato de que ambas não estão intimamente relacionadas ao estilo musical e representam pouco esforço de implementação, ao contrário da abordagem ImPact/*Cyber-João*, que requer um conhecimento profundo sobre o que deve ser tocado e como. Desta forma, o que se avalia, também, é até que ponto uma abordagem *knowledge intensive* (Russell 1995) é preferível em detrimento de outra menos sofisticada nesse sentido.

Devido à modelagem do sistema *Cyber-João*, a implementação dos dois novos sistemas necessitou de pouco esforço. No caso do *Crazy-João*, apenas um novo *seletor rítmico* precisou ser implementado. Na realidade, essa implementação consistiu apenas na mudança no acesso à memória musical, que, para o *Crazy-João*, é uma escolha aleatória em vez de *Motor de RBC* ou sequer um *Disparador de Regras*.

Para o *João-in-a-Box*, as modificações também exigiram poucos cuidados. Porém, algumas modificações foram feitas em relação às regras em si. O *Disparador de Regras* servirá, neste caso, como o filtro de padrões *a priori*.

Deve ficar claro que, em última instância, as regras do *Cyber-João* também filtram padrões. A grande diferença, entretanto, é que no *João-in-a-Box* essas regras agem diretamente na memória musical, retirando do processo de escolha os padrões que não estiverem em conformidade com elas. No *Cyber-João*, as regras refinam a consulta a partir da modificação dos valores dos atributos que um padrão deve possuir para que possa ser usado naquele momento.

As regras utilizadas no *João-in-a-Box* foram adaptadas daquelas implementadas para o *Cyber-João*. Porém, nem todas as regras do *Cyber-João*, podem efetivamente ser traduzidas a restrições diretas a casos. Por exemplo, para manter a coerência com o que foi tocado, o *Cyber-João* mantém uma regra que faz uma cópia dos valores dos atributos (apenas as propriedades musicais) do padrão recuperado anteriormente. Se traduzirmos isso para uma regra de restrição *a priori*, existe uma forte chance de restar apenas um padrão a ser escolhido: aquele recuperado anteriormente. Se alguma outra regra executada posteriormente causar a eliminação desse padrão também, nenhum padrão poderá ser recuperado para esse segmento.

Optamos, então, por usar regras que restrinjam grupos inteiros de padrões, como padrões de virada, de início, cíclicos e especiais. Na realidade, essa abordagem é a que deve ser usada idealmente, visto que é exatamente desta forma (restrição por grupos) que ocorre no *Band-in-a-Box*. Desta forma, quatro regras foram implementadas inspiradas naquelas existentes no *Cyber-João*. Essa regras podem ser vistas no Quadro 2.

<p><b>Regra: Início de Música</b></p> <p><b>SE</b></p> <p>(é o início da música)</p> <p><b>ENTÃO</b></p> <p>“elimine todos os padrões que não sejam de início“</p>
<p><b>Regra: Virada</b></p> <p><b>SE</b></p> <p>(melodia no último tempo do segundo compasso <math>\leq 2</math> notas) <b>E</b></p> <p>(se há um esquema V7-I entre o último acorde do segmento e o primeiro do próximo)</p> <p><b>ENTÃO</b></p> <p>“elimine todas os padrões que não sejam de virada“</p>
<p><b>Regra: Aplica padrões especiais</b></p> <p><b>SE</b></p> <p>(ritmo harmônico = 15)</p> <p><b>ENTÃO</b></p> <p>“Elimine todos os padrões que não sejam especiais“</p>

**Regra: Utilização Comum****SE**(não é início de música) **E**(não é virada) **E**(não ritmo harmônico  $< > 15$ )**ENTÃO**

“Elimine todos os padrões que não sejam de cíclicos”

Quadro 2 – Todas as regras utilizadas no *João-in-a-box*

No *João-in-a-Box*, o processo de escolha do padrão é feito da seguinte forma: para cada segmento, os padrões disponíveis, isto é, aqueles que não foram retirados pelo filtro de regras, são escolhidos de forma aleatória, onde cada um dos padrões tem chances diferentes de serem recuperados dependendo do valor do seu atributo *frequência de uso* (discutido na seção 4.4). Por exemplo, um padrão que possui valor quatro para o atributo citado tem duas vezes mais chance de ser recuperado que um padrão que possui valor dois para o mesmo atributo e quatro vezes, que um de valor um.

#### 5.4. Protocolo Experimental

Uma vez estabelecido o ambiente de teste, resta escolher as músicas a serem usadas na avaliação e os critérios de avaliação propriamente ditos. Quanto às músicas, utilizamos quatro canções do repertório clássico da Bossa Nova: *Chega de Saudade*, *Desafinado*, *Lígia* e *Insensatez*. A escolha delas obedeceu prioritariamente ao critério de disponibilidade da música no formato D’Accord Violão, uma vez que a demanda de trabalho para edição de músicas nesse formato ainda é bastante grande. Um outro fator de escolha importante foi o andamento. Foram escolhidas duas canções lentas (*Lígia* e *Insensatez*) e duas rápidas (*Chega de Saudade* e *Desafinado*).

Cada uma das músicas foi gerada a partir de cada um dos sistemas, resultando em um *corpus* de 12 músicas a serem avaliadas. Esse *corpus* foi apresentado a seis avaliadores, indicados no texto por nomes fictícios, cujo grau de instrução em música vem a seguir:

- O *avaliador1* é musicólogo e violonista com dezenas de anos de experiência. Especialista em Samba e Bossa Nova;
- o *avaliador2* é violonista profissional também com dezenas de anos de experiência em música erudita e popular;
- o *avaliador3* é violonista e baterista profissional, com grande experiência com Jazz e MPB;
- o *avaliador4* é pianista clássico profissional;
- o *avaliador5* é especialista em computação musical e músico amador de MPB;
- o *avaliador6* é guitarrista amador, tendo como preferências de estilo, o Funk e o Rock.

Para cada música, os avaliadores responderam ao questionário apresentado (ver Anexo C), no qual deveriam apontar, para cada uma das três amostras, suas notas para critérios como conformidade com o estilo e variabilidade, além da quantidade de erros por uso indevido dos padrões e da indicação da melhor e pior partes, caso seja perceptível. Por fim, cada avaliador foi convidado a escolher, para cada música, qual das amostras teve melhor desempenho geral.

Vale salientar que os avaliadores não sabiam de antemão qual dos sistemas gerou cada amostra, com isso tentamos diminuir o viés na avaliação. Os resultados gerais são apresentados em seguir

#### **5.4.1. Resultados da música *Desafinado***

Em *Desafinado*, a escolha do melhor resultado como aquele gerado pelo *Cyber-João* foi praticamente unânime. Os avaliadores um e dois não conseguiram escolher a melhor entre as performances de *Cyber-João* e *João-in-a-box*. Nesta música, exceto para o *Crazy-João*, poucos erros foram anotados. A maioria dos avaliadores consideraram as viradas como as melhores partes para *Cyber-João* e *João-in-a-box*, sendo o primeiro capaz de escolher melhores padrões de virada. A Tabela 6 a seguir resume os resultados obtidos nas avaliações.

<b>Avaliador</b>	<b><i>Crazy-João</i></b>	<b><i>João-in-a-box</i></b>	<b><i>Cyber-João</i></b>	<b>Melhor</b>
<i>Avaliador1</i>	Aceitável	Excelente	Excelente	<b>Empate</b>
<i>Avaliador2</i>	Aceitável	Bom	Bom	<b>Empate</b>
<i>Avaliador3</i>	Ruim	Aceitável	Aceitável	<b><i>Cyber-João</i></b>
<i>Avaliador4</i>	Ruim	Aceitável	Bom	<b><i>Cyber-João</i></b>
<i>Avaliador5</i>	Ruim	Bom	Excelente	<b><i>Cyber-João</i></b>
<i>Avaliador6</i>	Aceitável	Bom	Ruim	<b><i>João-in-a-box</i></b>

Tabela 6 – Resultados da avaliação das amostras de Desafinado

#### 5.4.2. Resultados da música *Chega de Saudade*

Nesta música, quatro dos seis avaliadores citaram, como melhor parte da execução do *Cyber-João*, a parte cuja letra é “ser milhões de abraços apertado assim”. Vale a pena notar que essa é uma parte onde ocorre um ritmo harmônico alto e, conseqüentemente, onde se devem usar padrões especiais. Ambos os sistemas *Cyber-João* e *João-in-a-box*, têm o conhecimento necessário para essa decisão. Porém, *João-in-a-box* escolhe, nesse caso, um padrão especial mais denso, o que entra em choque com uma melodia também densa. A Tabela 7 exhibe os resultados.

<b>Avaliador</b>	<b><i>Crazy-João</i></b>	<b><i>João-in-a-box</i></b>	<b><i>Cyber-João</i></b>	<b>Melhor</b>
<i>Avaliador1</i>	Aceitável	Excelente	Excelente	<b><i>Cyber-João</i></b>
<i>Avaliador2</i>	Ruim	Bom	Excelente	<b><i>Cyber-João</i></b>
<i>Avaliador3</i>	Aceitável	Excelente	Aceitável	<b><i>João-in-a-box</i></b>
<i>Avaliador4</i>	Ruim	Bom	Aceitável	<b><i>João-in-a-box</i></b>
<i>Avaliador5</i>	Ruim	Bom	Excelente	<b><i>Cyber-João</i></b>
<i>Avaliador6</i>	Aceitável	Aceitável	Bom	<b><i>Cyber-João</i></b>

Tabela 7 – Resultados da avaliação das amostras de *Chega de Saudade*

#### 5.4.3. Resultados de música *Insensatez*

Foi nesta música onde ocorreram os resultados mais inesperados. Três dos avaliadores consideraram as músicas geradas por *Cyber-João* e *Crazy-João* monótonas. O *avaliador6* chega a ponto de classificar a amostra gerada por *Crazy-João* como o melhor resultado para a música.

Um detalhe importante é que apenas os avaliadores que costumam tocar músicas de estilo mais dinâmico chegaram a citar a monotonia das músicas como um problema. Em contrapartida, o *avaliador1* cita que as Bossas lentas permitem uma melhor avaliação do emprego dos padrões rítmicos. Segundo ele: “nas [músicas] lentas, as mudanças de ritmo têm que ser feitas com mais cuidado”.

A Tabela 8 mostra os resultados obtidos nas avaliações.

<b>Avaliador</b>	<b><i>Crazy-João</i></b>	<b><i>João-in-a-box</i></b>	<b><i>Cyber-João</i></b>	<b>Melhor</b>
<i>Avaliador1</i>	Ruim	Excelente	Excelente	<b><i>Cyber-João</i></b>
<i>Avaliador2</i>	Aceitável	Bom	Bom	<b>Empate</b>
<i>Avaliador3</i>	Ruim	Ruim	Bom	<b><i>Cyber-João</i></b>
<i>Avaliador4</i>	Ruim	Ruim	Bom	<b><i>Cyber-João</i></b>
<i>Avaliador5</i>	Ruim	Aceitável	Excelente	<b><i>Cyber-João</i></b>
<i>Avaliador6</i>	Excelente	Bom	Aceitável	<b><i>Cyber-João</i></b>

Tabela 8 – Resultados da avaliação das amostras de *Insensatez* para cada avaliador

#### 5.4.4. Resultado da música *Lígia*

Mais uma vez, as críticas mais acentuadas foram por causa da monotonia (feitas novamente pelos mesmos avaliadores que criticaram a monotonia de *Insensatez*). Vale a pena perceber (ver Tabela 9) que o resultado geral da avaliação de *João-in-a-box* e *Cyber-João* cai sensivelmente em relação aos dados obtidos nas outras músicas (sobretudo para o *Cyber-João*, citado como o mais monótono dos três sistemas).

<b>Avaliador</b>	<b><i>Crazy-João</i></b>	<b><i>João-in-a-box</i></b>	<b><i>Cyber-João</i></b>	<b>Melhor</b>
<i>Avaliador1</i>	Aceitável	Bom	Excelente	<b><i>Cyber-João</i></b>
<i>Avaliador2</i>	Aceitável	Excelente	Excelente	<b><i>Cyber-João</i></b>
<i>Avaliador3</i>	Péssimo	Ruim	Péssimo	<b><i>João-in-a-box</i></b>
<i>Avaliador4</i>	Aceitável	Bom	Ruim	<b><i>João-in-a-box</i></b>
<i>Avaliador5</i>	Aceitável	Excelente	Excelente	<b><i>Cyber-João</i></b>
<i>Avaliador6</i>	Bom	Bom	Ruim	<b><i>João-in-a-box</i></b>

Tabela 9 – Resultados da avaliação das amostras de *Lígia* para cada avaliador

### 5.4.5. Resultados Gerais

Agrupando os resultados da avaliação de *melhor resultado* para cada um dos sistemas, obtemos a Tabela 10. Nela, é possível perceber que as amostras geradas pelo *Cyber-João* são citadas como melhor resultado numa frequência duas vezes maior que o *João-in-a-box*. É importante frisar que, em casos de empate, decidimos pontuar os dois sistemas empatados.

Em média, o *Cyber-João* e o *João-in-a-box* receberam uma nota geral “Bom”, enquanto o *Crazy-João* recebeu a nota “Razoável”.

Música	<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
<b>TOTAL</b>	<b>1</b>	<b>9</b>	<b>17</b>

Tabela 10 – Totalização do quesito melhor resultado

Em comentários gerais, o *Cyber-João* e o *João-in-a-box* foram considerados bastante satisfatórios. Principalmente o primeiro, que foi considerado mais “estável”, recebeu alguns comentários elogiosos. Ao ser questionado sobre o desempenho do sistema em relação ao de um ser humano, o *avaliador1* respondeu o seguinte:

*“[...] tirando a expressividade, se um aluno meu tocasse essas músicas dessa maneira para mim em busca de comentários, diria que ele havia aprendido a empregar os padrões rítmicos de Bossa Nova.”*

O mesmo avaliador, após a avaliação, ao ser esclarecido sobre qual sistema teria gerado cada música, chegou a dizer que os resultados estavam bastante coerentes:

*“Em Bossas rápidas há mais liberdade para variações, portanto é mais difícil escolher qual dos dois programas [Cyber-João e o João-in-a-box] é melhor, a não ser por alguns detalhes. Nas [músicas] lentas as mudanças de ritmo têm que ser feitas com mais cuidado. O Cyber-João, nesse caso, é muito melhor.”*

Um outro comentário bastante positivo foi proferido pelo *avaliador2*: “Em alguns casos, acho que um humano não faz escolhas melhores que ele [Cyber-João]”, citando principalmente o emprego do padrão usado na parte “... ser milhões de

abraços ...” de *Desafinado*. Ele continua: “[...] é exatamente assim que se deve fazer.”

### 5.5.Considerações Finais

Este capítulo apresentou os detalhes de implementação do *Cyber-João*, a criação do ambiente de avaliação, os outros dois sistemas criados como objetivo de comparação de resultados (*Crazy-João* e *João-in-a-box*) e os resultados obtidos a partir das respostas do questionário pelos avaliadores.

Embora os resultados sejam preliminares, podemos perceber uma grande superioridade dos sistemas mais elaborados (*Cyber-João* e *João-in-a-box*) em relação ao sistema totalmente aleatório, confirmando a suspeita inicial de que o emprego de conhecimento musical melhora significativamente o resultado do acompanhamento rítmico obtido.

Ainda de acordo com os resultados, foi possível perceber quem em determinados casos (canções lentas, no estilo Bossa Nova), o esforço de assumir um maior controle sobre o ambiente, representado pela abordagem *knowledge intensive* do *Cyber-João*, pode ser capaz de produzir resultados mais coerentes. Embora esse também fosse um resultado esperado, não prevíamos que o resultado fosse tão facilmente percebido, o que foi um resultado encorajador.



## 6. Conclusão

Este trabalho apresentou um estudo sobre a geração automática de acompanhamento rítmico para o violão. Embora a motivação inicial do trabalho tenha sido de suavizar a deficiência do D'Accord Violão na referida tarefa, esse problema é mais geral, sendo útil em qualquer programa que simule, de alguma forma, o acompanhamento rítmico de instrumentos. De fato, curiosamente, apesar de sua importância na implementação de Sistemas de Acompanhamento Automático e de Improvisação, pouca literatura existe na área de Computação Musical sobre ritmos, mais especificamente acompanhamento, rítmico.

Como estudo de caso, implementamos o *Cyber-João* um programa capaz de gerar acompanhamento rítmico de Bossa Nova a partir da grade de acordes a melodia da música sendo tocada. O sistema utiliza a abordagem proposta no sistema ImPact que combina raciocínio baseado em regras e em casos, contando atualmente com seis regras e 21 casos (padrões rítmicos e seus atributos). Para que pudéssemos fazer uso dessa abordagem foram necessários uma nova biblioteca de fragmentos musicais, novos atributos para descrever esses fragmentos, novas informações sobre o contexto musical (características da melodia e da grade de acordes que são relevantes) e novas regras para guiar a escolha de que fragmento usar.

Para auxiliar na avaliação dos resultados do *Cyber-João*, foram criados ainda dois sistemas de menor complexidade: O *Crazy-João*, que escolhe os padrões de forma totalmente aleatória, servindo como patamar mínimo de qualidade e o *João-in-a-Box*, que escolhe os padrões de forma aleatória, porém dando mais chances de recuperação aos padrões mais utilizados. O *João-in-a-Box* utiliza algumas das regras adaptadas a partir do *Cyber-João*, porém o faz de maneira um pouco diferente. Nele, essas regras agem efetivamente sobre os grupos de padrões funcionando como uma espécie de filtro.

Os resultados obtidos a partir de quatro músicas do repertório clássico da Bossa Nova (*Desafinado*, *Chega de Saudade*, *Insensatez* e *Lígia*) foram bastante satisfatórios. Os seis avaliadores indicaram resultados musicais entre aceitável e excelente para o *João-in-a-Box* e o *Cyber-João*, este último sendo mais “estável” musicalmente e, por isso, escolhido, em média, na grande maioria dos casos melhor sistema entre os três. Em alguns comentários, um dos avaliadores chegou a dizer que, em alguns casos, um humano não tem como fazer melhor que o *Cyber-João*.

Além disso, a equipe D’Accord Violão já utiliza o *Cyber-João* para simplificar o processo de edição de acompanhamento rítmico considerando o sistema “mais que satisfatório”.

### **6.1.Contribuições e Dificuldades Encontradas**

A principal contribuição desse trabalho, na nossa opinião, é prover de forma pioneira um estudo detalhado sobre geração automática de acompanhamentos rítmicos. Sem dúvida, as informações contidas neste trabalho podem ser generalizadas para qualquer instrumento de acompanhamento harmônico. A mão esquerda do pianista popular, por exemplo, freqüentemente, faz uma tarefa semelhante à feita pela mão direita do violão. Para casos como esse, muito do que foi dito aqui tem validade.

Uma contribuição mais específica para a geração automática de acompanhamento rítmico de Bossa Nova para o violão foi a catalogação dos padrões rítmicos específicos de viradas. Uma outra contribuição que pode ser apontada, ainda que secundária, é a criação de um motor RBC em C++ implementado tendo em vista reusabilidade e extensibilidade, podendo, então ser utilizado em qualquer outra aplicação, para qualquer fim.

Como principal dificuldade de ordem técnica, podemos citar a tarefa de estabelecer os atributos que descrevem os padrões, visto que a literatura musical disponível não trata dessas informações, dando apenas algumas “pistas” sobre o assunto.

A maior dificuldade de ordem prática está relacionada à análise qualitativa dos resultados, uma vez que não há uma metodologia consolidada para esse tipo de análise. Além disso, devido à necessidade da presença de seres humanos para avaliar as músicas geradas, fatores como, agendamento de horários e, muitas vezes, dificuldades associadas à comunicação entre profissionais de áreas distintas, acabaram atrasando o processo de avaliação.

## **6.2.Trabalhos Futuros e em Andamento**

Muito trabalho ainda deve ser feito para a melhora do sistema. A primeira tarefa necessária é testar os resultados com uma quantidade maior de músicas e de avaliadores, a fim de validar os resultados utilizando métodos estatísticos.

A integração do *Cyber-João* ao D'Accord Violão Editor não está completamente concluída, uma vez que o segundo passa por um processo de reformulação.

Faz-se necessário ainda que a abordagem adotada seja testada para outros estilos. Alguns passos já foram dados nesse caminho com alguns estudos sobre estilos como samba, frevo e baião. A partir dessa análise preliminar, percebe-se que os atributos e a maioria das regras estabelecidas para o *Cyber-João* não são específicos da Bossa Nova. Obviamente, um estudo mais aprofundado é necessário nesse sentido.

Como um outro trabalho adicional, poder-se-ia incluir fatores de expressividade no *Cyber-João*. Essa característica melhoraria sobremaneira a experiência musical obtida e ainda representaria avanço para o aprendizado de acompanhamento rítmico para o violão, necessidade de alguns dos usuários do D'Accord Violão.

É de grande valia, também, utilizar e avaliar outras técnicas de inteligência artificial, como por exemplo Redes Neurais, a fim de reforçar o trabalho de análise proposto.

Um outro trabalho importante, este já em andamento, é a generalização do modelo do *Cyber-João*. O objetivo dessa generalização é evitar o desgastante trabalho de procurar, catalogar e escolher os atributos que descreverão os fragmentos musicais, ou seja, o processo de aquisição do conhecimento.

Um programa que atendesse a esses requisitos deveria funcionar da seguinte forma (ver Figura 26 para o seu diagrama de funcionamento): O usuário, alimentaria o sistema com um *corpus* de música no estilo desejado. O programa então extrairia automaticamente os padrões (rítmicos, melódicos harmônicos etc.) relevantes, utilizando técnicas de aprendizagem de máquina (Mitchell 1997; Duda 2000), de acordo com as configurações estabelecidas pelo próprio usuário, como por exemplo, número máximo de padrões e os atributos sobre os quais a extração seria feita (adicionalmente, uma pequena linguagem poderia ser disponibilizada para que o próprio usuário fosse capaz de definir seus atributos a partir de outros atributos já existentes.) O resultado da extração, caso aprovado pelo usuário, poderia ser inserido diretamente na memória musical do *Cyber-João*, integrando a solução.

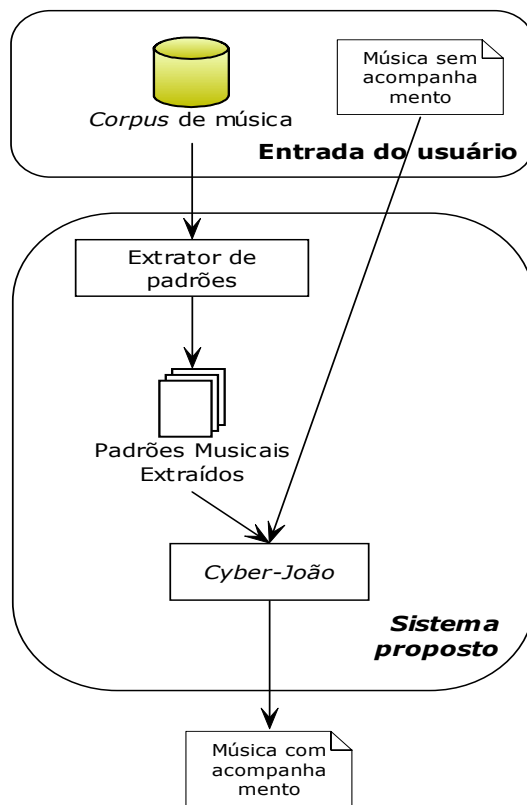


Figura 26 – Diagrama de funcionamento do programa proposto

Para realização desse sistema, já foi implementado, em parceria com o então aluno de graduação do CIn, Hugo Pimentel Santana, o *VexPat* (Santana 2003), uma ferramenta de extração e visualização de padrões musicais (Figura 27). Em linhas gerais, este sistema utiliza distância de edição (Manber 1989) como medida

de similaridade para implementar *matching* aproximado, mais especificamente, utiliza algumas operações de edição próprias para a música (Mongeau 1990; Rolland 2001).

O *VexPat* permite que o usuário navegue pelos padrões extraídos, indicado as semelhanças e diferenças entre os padrões.

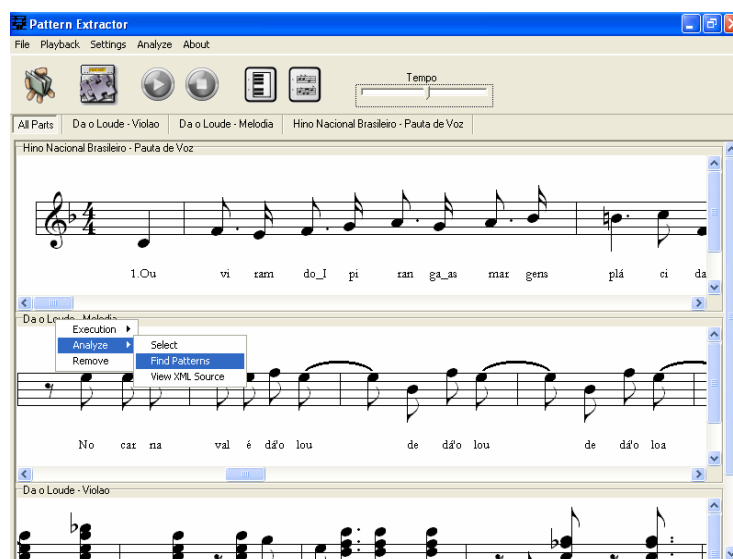


Figura 27 – Tela principal do *VexPat*

Os resultados obtidos a partir da extração de padrões nos formatos MIDI e MusicXML (Good 2002) encorajam o seu uso no programa proposto anteriormente. Estamos, atualmente, criando a integração entre os dois sistemas (*VexPat* e *Cyber-João*).

Decidimos não incluir este tema (extração automática de padrões) no documento da dissertação para mantermos o foco no processo de geração de acompanhamento rítmico, mas certamente a extração de padrões estende a contribuição dada durante o mestrado.

## 7. Referências Bibliográficas

- Aamodt, A., & Plaza, E. (1994). "Case-Based Reasoning: Fundamental Issues, Methodological Variations, and system Approaches." Artificial Intelligence Communications 7(1): 39-59.
- Abel, M. (2003). Raciocínio Baseado em Casos. Porto Alegre. 2004.
- Américo, L. (2004). A História da MPB.
- Ames, C., & Domino, M. (1992). *Cybernetic Composer: an overview*. Understanding Music with AI: Perspectives on Music Cognition. M. Balaban, Ebicioglu, K. & Laske, O. Menlo Park, The AAAI Press: 186-205.
- Arcos, J., Mántaras, R. & Serra, X. (1998). "SaxEx: A Case-Based Reasoning System for Generating Expressive Musical Performances." Journal of New Music Research 27(3): 194-210.
- 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.
- Baker, D. (1980). Miles Davis' Trumpet. Libano, Studio 224.
- Beck, S. (1996). "The Virtual Instrument Paradigm: An Alternative Approach to Interactive Computer Music." Computers & Mathematics with Applications 32(1): 5-10.
- Biles, J. (1994). A Genetic Algorithm for Generating Jazz Solos. International Computer Music Conference.
- Boden, M. (1992). The Creative Mind: Myths and Mechanisms. New York, Basic Books.
- Booch, G., Rumbaugh, J., & Jacobson, I. (1999). The Unified Modeling Language User Guide. Reading, Addison-Wesley.
- Cabral, G. (2002). D'Accord Guitar: Um Sistema para Execução Violonística. Centro de Informática. Recife, UFPE: 157.

- Cabral, G., et al (2001). "D'Accord Guitar: an Innovative Guitar Performance System." *The Journées d'Informatique Musicale*: 127 -137.
- Carroll, J., & Long, D. (1989). *Theory of Finite Automata*. Eaglewood Cliffs, Prentice-Hall.
- Chediak, A. (1986). *Harmonia e Improvisação*. Rio de Janeiro, Lumiar Editora.
- Clarke, E. (1993). "Imitating and Evaluating Real and Transformed Musical Performances." *Music Perception* **10**(3): 317 - 341.
- Cope, D. (1991). *Recombinant Music: Using the Computer to Explore Musical Style*. *IEEE Computer*. **24**: 22-28.
- Duda, R., Hart, P. & Stork, D (2000). *Pattern classification*. New York, John Wiley & Sons.
- Ebcioğlu, K. (1992). *An expert System for Harmonizing Chorales in the Style of J.S. Bach*. *Understanding Music with AI: Perspectives on Music Cognition*. M. Balaban, Ebcioğlu, K. & Laske, O. Menlon Park, The AAAI Press: 294-333.
- Garcia, W. (1999). *Bim Bom: A contradição sem conflitos de João Gilberto*. São Paulo, Paz e Terra.
- Gilberto, J. (1995). *Eu sei que eu vou te amar*, Sony Discos: Coletânea de músicas.
- Gilberto, J. (1998). *Joao Gilberto Prado Pereira de Oliveira*, wea: Coletânea de sucessos, ao vivo.
- Gilberto, J. (2000). *O Melhor da Música de João Gilberto*, wea: Coletânea de Sucessos.
- Giomi, F., & Ligabue, M. (1991). "Computational Generation and Study of Jazz Music." *Interface* **20**(1): 47-63.
- Good, M. (2002). *MusicXML in Practice: Issues in Translation and Analysis*. *International Conference MAX: Musical Application Using XML*.
- Hidaka, I., Goto, M. & Muraoka, Y. (1995). *An Automatic Jazz Accompaniment System Reacting to Solo*. *International Computer Music Conference, Banff, International Computer Music Association*.

- Hill, B. (1994). MIDI for Musicians: Buying, Installing, and Using Today's Electronic Music-Making Equipment. Atlanta, A Cappella.
- Johnson-Laird, P. (1991). *Jazz Improvisation: A Theory at the Computational Level. Representing Musical Structure*. P. Howell, West, R. & Cross, I., Academic Press.
- Keifer, B. (1984). Elementos da Linguagem Musical. Porto Alegre, Ed. Movimento.
- Kolodner, J. (1993). Case Based Reasoning. San Mateo, Morgan Kaufmann.
- Lerhdal, F., & Jackendoff, R. (1983). A Generative Theory of Tonal Music, The MIT Press.
- Levitt, D. (1983). *A Melody Description System for Jazz Improvisation*. Dept. of Electrical Engineering, Massachusetts Institute of Technology.
- Manber, U. (1989). Introduction to Algorithms: A Creative Approach, Addison-Wesley.
- McClain, E. (1994). *Musical Theory and Ancient Cosmology*. **2004**.
- Mitchell, M. (1996). An Introduction to Genetic Algorithms. Cambridge, The MIT Press.
- Mitchell, T. (1997). Machine Learning. New York, McGraw-Hill.
- Mongeau, M., & Sankoff, D. (1990). "Comparison of musical sequences." Computer and the humanities **24**: 161-175.
- Newell, A., & Simon, H. (1972). Human Problem-Solving, Prentice Hall.
- Owens, C. (1974). *Charlie Parker: Techniques of Improvisation*. Los Angeles, University of Los Angeles.
- Pachet, F. (1991). Representing Knowledge Used by Jazz Musicians. International Computer Music Conference, Montreal.
- Papadopoulos, G., & Wiggins, G. (1999). AI Methods for Algorithmic Composition: A Survey, a Critical View and Future Prospects. *Artificial Intelligence and the Simulation of Behaviour*, Edinburgh.
- Pereira, F., Grilo, C., Macedo, L. & Cardoso, A. (1997). Composing Music with Case-Based Reasoning. Second Conference on Computational Models of Creative Cognition, Dublin.



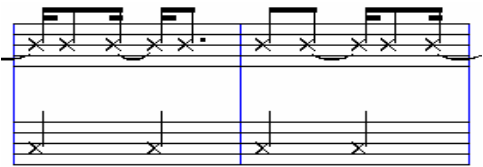
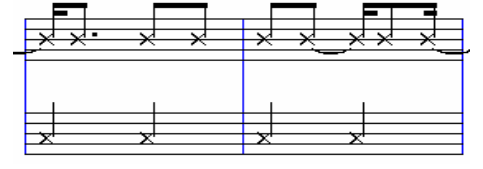
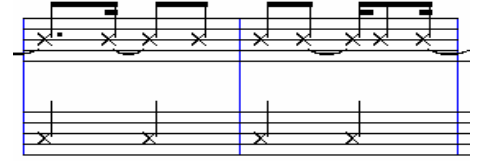
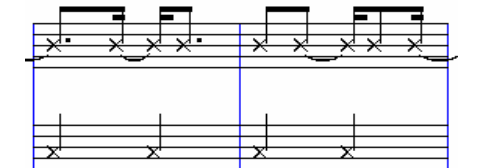
- Pressing, J. (1988). *Improvisation: Methods and Models. Generative Processes in Music: The Psychology of Performance, Improvisation and Composition*. J. Sloboda, Oxford Science Publications: 78-129.
- Ramalho, G. (1997). *Construction d'un Agent Rationnel Jouant du Jazz. Informatique*. Paris, Université Paris VI.
- Ramalho, G., Rolland, P-Y. & Ganascia, J-G. (1999). "An Artificially Intelligent Jazz Performer." *Journal of New Music Research* **28**(2): 105-129.
- Ramalho, G. G., J-G. (1994). *Simulating Creativity in Jazz Performance*. Twelfth National Conference on Artificial Intelligence, Seattle, The AAAI Press.
- Roads, C. (1995). *The Computer Music Tutorial*. Cambridge, MIT Press.
- Rolland, P.-Y. (2001). *FLExPat: Flexible Extraction of Sequential Patterns*. The IEEE International Conference on Data Mining, The IEEE Press.
- Rolland, P.-Y., & Ganascia, J-G. (1996). *Automated Extraction of Prominent Motives in Jazz Solo Corpuses*. 4th International Conference on Music Perception and Cognition, Montreal, McGill University.
- Rothstein, J. (1992). *MIDI: A Comprehensive Introduction*, A-R Editions.
- Rowe, R. (1993). *Interactive Music Systems*. Massachusetts, The MIT Press.
- Russell, S., & Norvig, P. (1995). *Artificial Intelligence: A Modern Approach*. Englewood Cliffs, Prentice Hall.
- Sabatella, M. (1996). *A Jazz Improvisation Primer*. **2004**.
- Sabater, J., Arcos, J. & Mántaras, R. (1998). *Using rules to support case-based reasoning for harmonizing melodies*. The AAAI Spring Symposium Multimodal Reasoning, California, The AAAI Press.
- Sandroni, C. (1985). O Olhar do Aprendiz: Observações sobre a prática do violão popular no Brasil, fartamente documentada por exemplos em partitura. (artigo não publicado).

- Santana, H., Dahia, M., Trajano, E. & Ramalho, G. (2003). VexPat - An Analysis Tool for the Discovery of Musical Patterns. IX Brazilian Symposium on Computer Music, Campinas - SP.
- Schank, R. (1982). Dynamic Memory: A Theory of Learning in Computers and People. Cambridge, Cambridge University Press.
- Schönberg, A. (1922). Harmonielehre, Universal Edition.
- Schurmann, E. (1989). A Música como Linguagem: uma Abordagem Histórica. São Paulo, Editora Brasiliense.
- Sommerville, I. (2000). Software Engineering, Addison Wesley.
- Spector, L., & Alpern, A. (1994). Criticism, Culture and the Automatic Generation of Artworks. Twelfth National Conference on Artificial Intelligence, The AAAI Press.
- Steedman, M. (1984). "A Generative Grammar for Jazz Chord Sequences." Music Perception 1(2): 52-77.
- Strube, G., Enzinger, A., Janetzko, D., & Knauff, M. (1995). Knowledge Engineering for CBR systems from a Cognitive Science Perspective. First International Conference on Case-Based Reasoning, Lisboa, Springer-Verlag.
- Trajano, E., Guigue, D., Costa, E., Silva, K., Gomes, C., Almeida, H. & Cavalcanti, N. (2003). SOS: A Tool for the Automatic Segmentation of Musical Flows. IX Brazilian Symposium on Computer Music, Campinas - SP.
- Walker, W. (1994). A Conversation-Based Framework for Musical Improvisation. Computer Science. Urbana, University of Illinois at Urbana-Champaign.
- Widmer, G. (1995). "Modeling the Rational Basis of Musical Expression." Computer Music Journal 19(2): 76-96.
- Widmer, G. (2001). "Using AI and Machine Learning to Study Expressive Music Performance: Project Survey and First Report." AI Communications 14(3): 149-162.

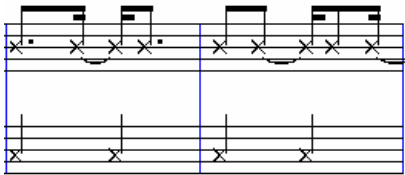


### Padrões Cíclicos

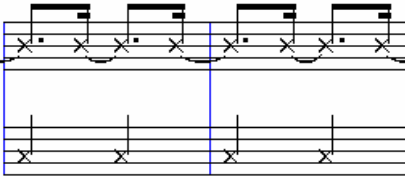

Padrão Rítmico	Andamento	Ritmo Harmônico	Densidade	Freq.de Uso	Início	Virada
	Lento ou Rápido	5 (1°. e 3°. tempos)	Média	5	Não	Não
	Lento ou Rápido	5 (1°. e 3°. tempos)	Baixa	4	Não	Não
	Rápido	5 (1°. e 3°. tempos)	Média	3	Não	Não

	Rápido	7 (1º, 2º. e 3º. tempos)	Média	3	Não	Não
	Rápido	7 (1º, 2º. e 3º. tempos)	Média	4	Não	Não
	Rápido	5 (1º. e 2º. tempos)	Baixa	2	Não	Não
	Rápido	6 (2º. e 3º. tempos)	Baixa	2	Não	Não

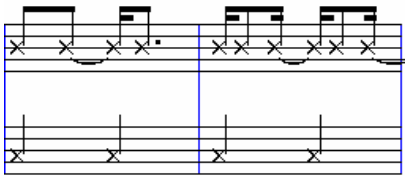
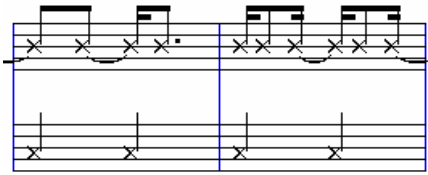
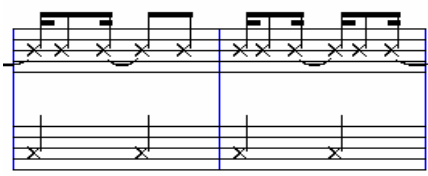


	Rápido	5 (1º. e 2º. tempos)	Baixa	2	Sim	Não
---	--------	----------------------	-------	---	-----	-----

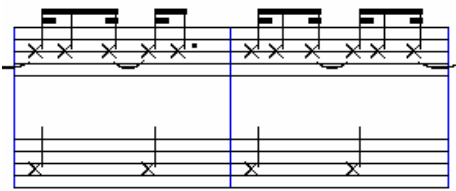
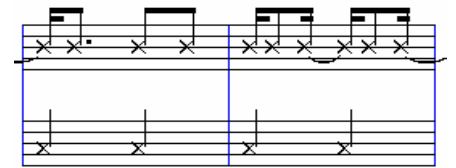
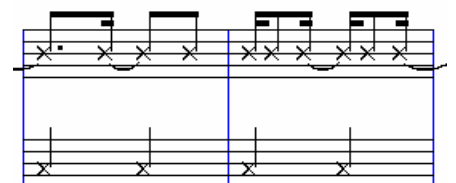

### Padrões Especiais

Padrão Rítmico	Andamento	Ritmo Harmônico	Densidade	Freq.de Uso	Início	Virada
	Rápido	15 (1º, 2º;3º e 4º tempos)	Baixa	3	Não	Não
	Rápido	15 (1º, 2º;3º e 4º tempos)	Alta	2	Não	Não

### Padrões de Virada

Padrão Rítmico	Andamento	Ritmo Harmônico	Densidade	Freq.de Uso	Início	Virada
	Rápido	5 (1°. e 2°. tempos)	Baixa	1	Não	Sim
	Rápido	5 (1°. e 2°. tempos)	Baixa	3	Não	Sim
	Rápido	5 (1°. e 2°. tempos)	Média	1	Não	Sim



	Rápido	7 (1º, 2º. e 3º. tempos)	Média	2	Não	Sim
	Rápido	7 (1º, 2º. e 3º. tempos)	Média	3	Não	Sim
	Rápido	5 (1º. e 2º. tempos)	Baixa	1	Não	Sim
	Rápido	6 (2º. e 3º. tempos)	Baixa	2	Não	Sim

## Anexo B – Lista das Regras

Este anexo apresenta a lista completa das seis regras disponíveis ao processo de decisão do *Cyber-João*. Vale a pena esclarecer alguns pontos. Em alguns momentos, no conseqüente das regras aparecem frases como “use as propriedades musicais do padrão anterior”. Essa frase está substituindo a atribuição de todas as propriedades musicais de um padrão para o outro. Há também a frase “use as propriedades musicais diferentes do padrão anterior”, neste caso o ocorre é atribuição de propriedades musicais contrastantes com as do padrão anterior. Por exemplo, se o padrão anterior possui densidade alta, o padrão a ser recuperado deve possuir densidade baixa. No caso da densidade média, o padrão a ser recuperado também deve possuir essa densidade.

### Manipula Início da Música

**SE**

(é o início da música)

**ENTÃO**

início = sim;

densidade = baixa;

### Aplica virada

**SE**

(melodia no último tempo do segundo compasso  $\leq 2$  notas) **E**

(se há um esquema V7-I entre o último acorde do segmento e o primeiro do próximo)

**ENTÃO**

“use as propriedades musicais do padrão anterior”

virada = sim;

### Mantém coerência com a melodia 1

**SE**

(densidade melódica  $\geq 10$  notas no segmento)

**ENTÃO**

densidade = baixa

## Mantém coerência com a melodia 2

### SE

(densidade melódica  $\leq 5$  notas no segmento)

### ENTÃO

densidade = baixa

## Mantém coerência com o que já foi tocado

### SE

(contexto atual  $\sim$  contexto anterior)

### ENTÃO

“use as propriedades musicais do padrão anterior”

## Faz variação em relação ao que já foi tocado

### SE

(contexto atual  $\neq$  contexto anterior) E

(tocando o mesmo padrão há 6 segmentos)

### ENTÃO

“use as outras propriedades musicais diferentes do padrão anterior”

frequência de uso = baixa;

## Anexo C – Questionário de Avaliação dos Sistemas

Questionário apresentado aos avaliadores do sistema.

### QUESTIONÁRIO DE AVALIAÇÃO

**Música:** \_\_\_\_\_

#### Amostra 1

##### Nota Geral:

☐ Péssimo    ☐ Ruim    ☐ Aceitável    ☐ Bom    ☐ Excelente

##### Conformidade com o Estilo:

☐ Não é Bossa Nova    ☐ É Bossa em alguns momentos    ☐ É Bossa Nova

##### Variabilidade:

☐ Monótona    ☐ Balanceada    ☐ Exagerada

**Total de erros (uso indevido de padrão) encontrados:** \_\_\_\_\_

**Melhor Parte:** \_\_\_\_\_

**Pior Parte:** \_\_\_\_\_

#### Amostra 2

##### Nota Geral:

☐ Péssimo    ☐ Ruim    ☐ Aceitável    ☐ Bom    ☐ Excelente

##### Conformidade com o Estilo:

☐ Não é Bossa Nova    ☐ É Bossa em alguns momentos    ☐ É Bossa Nova

##### Variabilidade:

☐ Monótona    ☐ Balanceada    ☐ Exagerada

**Total de erros (uso indevido de padrão) encontrados:** \_\_\_\_\_

**Melhor Parte:** \_\_\_\_\_

**Pior Parte:** \_\_\_\_\_

#### Amostra 3

##### Nota Geral:

☐ Péssimo    ☐ Ruim    ☐ Aceitável    ☐ Bom    ☐ Excelente

##### Conformidade com o Estilo:

☐ Não é Bossa Nova    ☐ É Bossa em alguns momentos    ☐ É Bossa Nova

##### Variabilidade:

☐ Monótona    ☐ Balanceada    ☐ Exagerada

**Total de erros (uso indevido de padrão) encontrados:** \_\_\_\_\_

**Melhor Parte:** \_\_\_\_\_

**Pior Parte:** \_\_\_\_\_

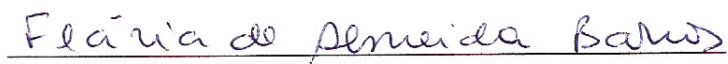
**Melhor Resultado:**

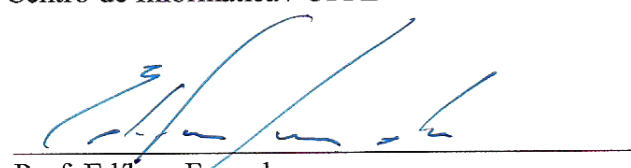
☐ Amostra 1

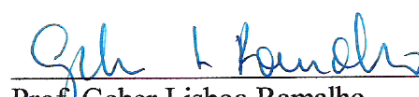
☐ Amostra 2

☐ Amostra 3

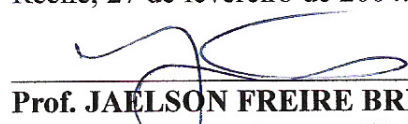
Dissertação de Mestrado apresentada por **Marcio Leal de Melo Dahia** a Pós-Graduação em Ciência da Computação do Centro de Informática da Universidade Federal de Pernambuco, sob o título , **“Gerando Acompanhamento Rítmico Automático para Violão: Estudo de caso do Cyber-João”**, orientada pelo **Prof. Geber Lisboa Ramalho** e aprovada pela Banca Examinadora formada pelos professores:

  
\_\_\_\_\_  
Prof. Flavia de Almeida Barros  
Centro de Informática / UFPE

  
\_\_\_\_\_  
Prof. Edilson Farneda  
Pró-Reitoria de Pesquisa e Pós-Graduação / UCB

  
\_\_\_\_\_  
Prof. Geber Lisboa Ramalho  
Centro de Informática / UFPE

Visto e permitida a impressão.  
Recife, 27 de fevereiro de 2004.

  
\_\_\_\_\_  
**Prof. JAELESON FREIRE BRELAZ DE CASTRO**  
Coordenador da Pós-Graduação em Ciência da Computação do  
Centro de Informática da Universidade Federal de Pernambuco.