



UNIVERSIDADE FEDERAL DE PERNAMBUCO  
MESTRADO EM CIÊNCIA DA COMPUTAÇÃO  
CENTRO DE INFORMÁTICA



# CINBALADA: UM LABORATÓRIO RÍTMICO

---

DISSERTAÇÃO DE MESTRADO

**Aluno:** Pablo Azevedo Sampaio (pas@cin.ufpe.br)  
**Orientadora:** Patrícia Cabral de Azevedo Restelli Tedesco (pcart@cin.ufpe.br)  
**Co-Orientador:** Geber Lisboa Ramalho (glr@cin.ufpe.br)

AGOSTO DE 2006

*"The mathematics of rhythm are universal.  
They don't belong to any particular culture."*

**(John McLaughlin)**

## Abstract

Artificial Intelligence research field has a long history of cooperation with Computer Music researches. Most of the works produced by such cooperation has dealt with the problem of automatic composition of musical elements. These works have focused mainly on the generation of harmony and melody, while the composition of rhythmic elements has been more rarely studied.

In the current work we present the CInBalada system for polyphonic rhythmic composition. The system is based on intelligent percussionist agents playing together in a group performance. The system has a database of pre-recorded monophonic rhythm patterns that the agents try to combine in interesting polyphonic performances by means of negotiation. To organise interactions between the agents, we have created the concept of *rhythmic role*, which represents the purpose that a percussion instrument has in a performance.

A *rhythmic role* defines some criteria for the selection of rhythm patterns played by an instrument. These criteria are used to evaluate not only individual properties of the rhythm patterns but also their suitability to the performance context. The criteria of the system's *rhythmic roles* were implemented using some measures of analysis of rhythm patterns found in the literature as well as other measures proposed here.

At the end of this work we present the results of an experiment which show that the CInBalada system has achieved its goals.

## Resumo

A Inteligência Artificial tem um longo histórico de colaboração com as pesquisas em Computação Musical. Dos trabalhos produzidos nessa interseção entre as duas áreas, a maior parte se voltou para o problema da composição automática de elementos musicais. Esses trabalhos historicamente deram preferência para a geração de harmonia e melodia, enquanto a composição de elementos rítmicos percussivos é pouco tratada na literatura.

Neste trabalho, apresentamos o sistema de composição rítmica polifônica CInBalada. Este sistema é composto por agentes percussionistas inteligentes tocando em conjunto. O sistema utiliza uma base de padrões rítmicos monofônicos pré-gravados que os agentes, por meio de negociação, buscam combinar para criar performances polifônicas diversas. Para organizar as interações entre os agentes, propusemos aqui o conceito de *papel rítmico*, que é a função exercida por um instrumento de percussão em uma performance.

Um *papel rítmico* define diversos critérios de seleção dos padrões rítmicos de um instrumento. Os critérios controlam tanto propriedades individuais dos padrões como a adequação destes ao contexto da performance. Para implementar os critérios definidos pelos papéis rítmicos, foram utilizadas diversas medidas de análise e comparação de padrões rítmicos encontradas na literatura, além de outras medidas que propomos aqui.

No final deste trabalho, apresentamos os resultados de um experimento onde mostramos que o CInBalada conseguiu atingir os objetivos propostos.

## **Agradecimentos**

Aos meus amados pais, Ivaldo e Valdeluce.

Aos meus orientadores, Patrícia e Geber.

Aos colegas de turma que estão comigo desde a graduação.

E ao meu bom Deus!

# Índice Analítico

<b>1. Introdução .....</b>	<b>1</b>
1.1 Objetivos .....	2
1.2 Metodologia .....	2
1.3 Estrutura da Dissertação .....	3
<b>2. Sistemas de Composição Automática .....</b>	<b>4</b>
2.1 Sistemas Baseados em Algoritmos Genéticos .....	5
2.2 Sistemas Baseados em Agentes .....	7
2.3 Considerações Finais .....	15
<b>3. Representação e Avaliação Rítmicas.....</b>	<b>16</b>
3.1 Estrutura Rítmica .....	16
3.2 Representações Musicais .....	22
3.3 Extração de Propriedades Rítmicas .....	29
3.4 Considerações Finais .....	36
<b>4. Framework para Aplicações Musicais .....</b>	<b>37</b>
4.1 Framework Multiagente .....	38
4.2 Agente Mixer .....	43
4.3 Considerações Finais .....	48
<b>5. Modelagem do CInBalada.....</b>	<b>49</b>
5.1 Abordagens Preliminares .....	50
5.2 Papéis Rítmicos.....	51
5.3 Arquitetura do Sistema .....	56
5.4 Base de Padrões Rítmicos .....	60
5.5 Representação da Estrutura Rítmica .....	62
5.6 Mecanismo de Avaliação .....	64
5.7 Negociação entre Agentes Percussionistas .....	71
5.8 Considerações Finais .....	74

<b>6. Implementação do CInBalada .....</b>	<b>76</b>
6.1 Padrões Rítmicos .....	76
6.2 Papéis Rítmicos.....	78
6.3 Agentes Percussionistas .....	80
6.4 Fachada .....	81
6.5 Interfaces.....	82
6.5 Considerações Finais .....	84
<b>7. Avaliação do Sistema .....</b>	<b>85</b>
7.1 Avaliação de Qualidade .....	85
7.2 Avaliação da Diversidade .....	90
7.3 Considerações Finais .....	94
<b>8. Conclusão.....</b>	<b>95</b>
8.1 Contribuições Principais e Lições Aprendidas .....	96
8.2 Trabalhos Futuros .....	97
<b>Bibliografia .....</b>	<b>101</b>
<b>Apêndice A: Questionário Sobre o Participante .....</b>	<b>109</b>
<b>Apêndice B: Questionário do Experimento de Qualidade .....</b>	<b>110</b>
<b>Apêndice C: Questionários do Experimento de Diversidade.....</b>	<b>111</b>

# Lista de Figuras

Figura 3.1: Batidas (pulsos maiores) e grade isocrônica (pulsos atômicos) .....	18
Figura 3.2: Três diferentes esquemas de pulsos: uma métrica regular, uma métrica complexa e um ritmo clave caribenho .....	19
Figura 3.3: Polirritmo 4:6 (quatro contra seis).....	20
Figura 3.4: Agrupamentos hierárquicos (em motivos e frases) e relações entre eles ..	21
Figura 3.5: Notações alternativas para o ritmo-clave de Rumba .....	22
Figura 3.6: Notação TUBS e variantes .....	23
Figura 3.7: Representação poligonal do ritmo clave de Rumba .....	24
Figura 3.8: Histograma de intervalos entre ataques do ritmo-clave de Rumba.....	24
Figura 3.9: Ritmo clave de Rumba na representação TEDAS.....	25
Figura 3.10: Histograma de frequências das distâncias entre ataques .....	25
Figura 3.11: Curva de andamento .....	27
Figura 3.12: Estrutura métrica para uma métrica quaternária regular .....	27
Figura 3.13: Representação cíclica de um métrica quaternária regular .....	28
Figura 3.14: Duas métricas possíveis em uma grade de nove pulsos .....	28
Figura 3.15: Padrão rítmico africano Shiko (Toussaint, 2002).....	30
Figura 3.16: Três classes de ritmos africanos de ciclos de doze unidades (Toussaint, 2003b) .....	30
Figura 3.17: Igualdade de espaçamento por regressão linear .....	32
Figura 3.18: Área mínima entre dois padrões rítmicos (doze, neste caso) .....	32
Figura 3.19: Cálculo da distância cronotônica usando os vetores cronotônicos.....	33
Figura 3.20: Padrão rítmico (a), usados pelos pigmeus africanos; padrão rítmico (b), usado no flamenco espanhol .....	33
Figura 3.21: Posições <i>off-beat</i> , segundo Toussaint.....	35
Figura 4.1: <i>Container</i> da plataforma FAMA .....	39
Figura 4.2: Diagrama de classes do FAMA.....	42
Figura 4.3: Escalonamento de <i>behaviours</i> no FAMA e no JADE.....	43
Figura 4.4: Exemplo de trecho musical em <i>fama-music-xml</i> .....	46
Figura 4.5: Classes que compõem o agente <i>mixer</i> .....	47

Figura 5.1: Partituras de alguns instrumentos de uma batucada de samba (Salazar, 1991) .....	52
Figura 5.2: Arquitetura do sistema CInBalada .....	57
Figura 5.3: Modelagem interna do agente percussionista.....	58
Figura 5.4: Representação de uma métrica quaternária complexa no CInBalada .....	63
Figura 5.5: Andamento no CInBalada .....	63
Figura 5.6: Medida de <i>interplay</i> – encontros e desencontros entre ataques .....	66
Figura 5.7: Padrão Shiko (tracejado) e padrão derivado (traço contínuo).....	67
Figura 5.8: Máquina de estados do protocolo de negociação .....	72
Figura 6.1: Classes que representam e armazenam os padrões rítmicos .....	77
Figura 6.2: Entrada XML para um padrão rítmico e um instrumento .....	78
Figura 6.3: Classes que representam os papéis rítmicos.....	79
Figura 6.4: Classes que compõem o agente percussionista .....	80
Figura 6.5: Fachada do sistema CInBalada.....	81
Figura 6.6: Interface principal do CInBalada .....	83
Figura 6.7: Editor de Repertório do CInBalada .....	84

# Lista de Tabelas

Tabela 2.1: Características dos <i>frameworks</i> .....	11
Tabela 3.1: Andamento real para algumas indicações em italiano .....	26
Tabela 4.1: Algumas performativas implementadas no FAMA .....	40
Tabela 4.2: Campos de mensagem implementados no FAMA.....	41
Tabela 4.3: Mensagens usadas pelo agente <i>mixer</i> .....	46
Tabela 5.1: Modelagem dos papéis rítmicos do CInBalada .....	70
Tabela 7.1: Orquestração em cada rodada do experimento de qualidade.....	87
Tabela 7.2: Experimento de qualidade – geral.....	88
Tabela 7.3: Experimento de qualidade – grupo dos músicos.....	89
Tabela 7.4: Experimento de qualidade – grupo dos não-músicos .....	89
Tabela 7.5: Experimento de diversidade – geral.....	93
Tabela 7.6: Experimento de diversidade – resultado separado por grupo .....	93

## CAPÍTULO 1

# Introdução

A música é tradicionalmente dividida em três componentes (ou dimensões): harmonia, melodia e ritmo. O *ritmo* corresponde à dimensão temporal da música e tem grande impacto na sensação causada pela música em ouvintes humanos (Honning, 2002). Provavelmente essa era a única dimensão presente nas primeiras manifestações musicais humanas e, mesmo nos dias de hoje, ela ainda é a única dimensão (ou a mais importante) em diversas manifestações musicais. Exemplos contemporâneos são: a música tribal africana (Ladzekpo, n.d.), as batucadas de Samba brasileiras (Salazar, 1991) ou mesmo gêneros mais modernos como Rap ou Disco. Com base nisso, Pazos et al. (1999) afirma que o ritmo é a única dimensão essencial para a existência de música.

Apesar da reconhecida importância da dimensão rítmica, os trabalhos da área de Inteligência Artificial aplicada à Computação Musical, em sua maioria, se preocupam primordialmente com melodia e harmonia. Sistemas de composição rítmica e sistemas interativos de acompanhamento percussivo, por exemplo, são problemas pouco tratados na literatura. Dentre esses poucos sistemas rítmicos encontrados na literatura, nós identificamos duas técnicas de Inteligência Artificial (IA) utilizadas: algoritmos genéticos (Mitchell, 1996) e o paradigma multiagente (Wooldridge, 2002). Comparando as duas técnicas, acreditamos que a abordagem baseada no paradigma multiagente apresenta algumas vantagens. A primeira vantagem é permitir uma modelagem mais natural de um grupo de músicos tocando em conjunto. Outra vantagem é que os agentes são capazes de produzir saídas em tempo real, enquanto os sistemas baseados em algoritmos genéticos requerem execuções relativamente longas.

Apenas dois dos sistemas de composição rítmica encontrados na literatura são baseados no paradigma multiagentes. Dos dois sistemas, apenas um modelou o processo de composição de maneira realmente distribuída pela interação entre agentes inteligentes (enquanto o outro sistema se preocupou apenas com a modelagem de um único agente). Porém, o sistema em questão não fechou todas as possíveis abordagens para a criação de composições rítmicas pela interação entre os agentes, deixando oportunidades a serem

exploradas. Neste trabalho, apresentamos o sistema multiagente CInBalada<sup>1</sup>, desenvolvido usando uma abordagem alternativa para o problema da composição.

Os agentes do sistema CInBalada agem como percussionistas tocando em conjunto. Os agentes dispõem de uma base de padrões rítmicos monofônicos pré-gravados que eles tentam combinar para gerar uma performance polifônica musicalmente aceitável. Para isso, os agentes usam negociação multiagente e um mecanismo descentralizado de avaliação dos padrões.

## **1.1 Objetivos**

O objetivo básico do sistema é compor arranjos rítmicos polifônicos com *qualidade musical aceitável*. Ou seja, o sistema precisa gerar saídas musicais que agradem a ouvintes humanos, especialmente músicos. Outro objetivo básico do sistema é gerar uma razoável *diversidade de composições*.

Além desses dois objetivos, esperamos que o sistema possa servir como *ferramenta de auxílio para músicos humanos*. Esse objetivo depende não apenas dos dois objetivos anteriores como também requer que o sistema ofereça aos usuários parâmetros musicais de alto-nível para personalização das composições.

## **1.2 Metodologia**

Para atingir os objetivos acima delineados foram seguidos os seguintes passos de pesquisa (não necessariamente em ordem cronológica):

1. **Levantamento bibliográfico de sistemas semelhantes.** Para contextualizar o sistema, foram pesquisados os sistemas de composição rítmica anteriores e as técnicas usadas por eles. Nessa pesquisa, foram encontradas as oportunidades de trabalhos originais deixadas.
2. **Levantamento bibliográfico dos elementos básicos que compõem a percepção rítmica humana.** Foram identificadas técnicas de representação desses elementos e medidas de avaliação que operam sobre padrões rítmicos. Esses elementos foram usados no sistema de modo a tornar as suas saídas mais relevantes para a percepção humana.

---

<sup>1</sup> O nome do sistema vem da junção da sigla do Centro de Informática da UFPE com o nome de um grupo de percussão brasileiro relativamente famoso nos anos 90.

3. **Abordagens preliminares de modelagem do sistema.** Serviram para validar a proposta de combinar padrões monofônicos prontos por meio de negociação multiagente. Foi identificada a necessidade de usar algum modelo conceitual para organizar as performances dos agentes.
4. **Elaboração de um modelo conceitual das performances rítmicas.** Proposta diante da escassez de trabalhos que tratem da estrutura de larga-escala das performance rítmicas polifônicas.
5. **Elaboração da modelagem final do sistema.** Usa o modelo proposto, de modo a superar as limitações encontradas nas abordagens preliminares. Feita com base nos requisitos de qualidade e diversidade das saídas rítmicas.
6. **Criação da base de padrões.** Para o sistema operar, foi montada uma base de padrões rítmicos encontrados em fontes diversas: em artigos da literatura, em ferramentas musicais e em sites da internet.
7. **Avaliação do sistema.** O sistema foi avaliado por meio de um experimento com humanos (divididos em músicos e não-músicos) quanto aos requisitos de *qualidade e diversidade*.

### **1.3 Estrutura da Dissertação**

No restante da dissertação apresentamos o trabalho de pesquisa introduzido neste capítulo. A contextualização do sistema proposto é feita no capítulo 2 desta dissertação, que apresenta um levantamento dos sistemas compositores rítmicos. Já no capítulo 3, apresentamos o levantamento bibliográfico de técnicas de representação e de avaliação de padrões rítmicos. Em seguida, no capítulo 4, apresentamos o *framework* para aplicações musicais desenvolvido para servir de ponto de partida para o sistema CInBalada. No capítulo 5 apresentamos a modelagem do CInBalada com base no conceito de *papéis rítmicos*, enquanto no capítulo 6 apresentamos a implementação feita a partir dessa modelagem. No capítulo 7, avaliamos o sucesso do CInBalada quanto ao cumprimento dos objetivos propostos. Por fim, no capítulo 8, discutimos os resultados deste trabalho e propomos novas linhas de pesquisa que podem ser seguidas a partir dele.

## CAPÍTULO 2

# Sistemas de Composição Automática

A área de Inteligência Artificial (IA) desempenha um papel crucial nas pesquisas em Computação Musical desde o início dessas duas áreas nos anos cinquenta, segundo Mantaras & Arcos (2002). Porém, nos anos oitenta é que a interação entre as áreas se tornou mais produtiva, com a disseminação do áudio digital e a adoção de um protocolo universal de controle de performances musicais (o protocolo MIDI), segundo Pressing (1997b).

Pearce (2000) afirma que as motivações para a aplicação de técnicas de IA ao domínio musical podem se situar desde a Ciência Cognitiva, em um extremo, à Engenharia, em outro extremo. No primeiro caso, o objetivo é modelar e entender características da cognição musical humana, tais como: percepção da música, aprendizado e aquisição de conhecimento musical, composição e criatividade musical. No extremo da Engenharia, os estudos visam produzir sistemas inteligentes que auxiliem ou enriqueçam os processos de aprendizado, composição e performance musicais.

Dentro dessa perspectiva da Engenharia, especial ênfase tem sido dada, historicamente, aos sistemas compositores, que, inspirados em Papadopoulos & Wiggins (1999), definimos aqui como “*sistemas capazes de resolver o problema de combinar ‘partes’ musicais (tais como: notas, frases e acordes) em um ‘todo’, que pode ser uma peça musical completa ou apenas um elemento dela (como: harmonia, melodia ou o arranjo de uma linha instrumental)*”. Por esta definição, uma vasta gama dos sistemas musicais inteligentes pode ser considerada um sistema compositor – não somente aqueles que geram uma peça final completa, mas todo sistema que cria resultados sonoros com os quais não foi alimentado.

Observamos que as pesquisas da área de sistemas de composição têm preterido a criação de sistemas de saídas puramente rítmicas (instrumentos percussivos) em favor de sistemas que gerem melodia ou harmonia (instrumentos de notas de alturas definidas). O

objetivo deste capítulo é analisar as técnicas utilizadas na construção dos sistemas rítmicos propostos na literatura.

N a primeira seção deste capítulo, analisamos o uso de algoritmos genéticos em sistemas de composição rítmica. Já na segunda seção, analisamos o emprego de agentes inteligentes para composição, dando especial ênfase aos dois sistemas rítmicos encontrados.

## **2.1 Sistemas Baseados em Algoritmos Genéticos**

Iniciaremos esta seção apresentando uma introdução aos algoritmos genéticos, na primeira subseção a seguir. Já na subseção seguinte, apresentamos uma análise dos sistemas de composição rítmica baseados nessa técnica.

### **2.1.1 Introdução aos Algoritmos Genéticos**

Os algoritmos genéticos formam uma família de técnicas inspiradas na Teoria da Evolução que fornecem soluções aproximadas para problemas de busca ou otimização. Analogamente ao mecanismo biológico de evolução, os algoritmos genéticos evoluem soluções (parciais) em sucessivas iterações (gerações). As soluções parciais, também chamadas de *indivíduos*, são representadas abstratamente por alguma estrutura computacional à qual se dá o nome de *cromossomos*.

Inicialmente, um algoritmo genético gera uma população inicial de indivíduos, o que geralmente acontece atribuindo-se valores randômicos aos cromossomos dos indivíduos. Em seguida, dá-se início ao processo de evolução, que produz sucessivas *gerações* de soluções parciais. As novas gerações são produzidas pela aplicação das seguintes classes de operadores:

- **Mutação:** alteram alguns dos cromossomos de uma solução. Inspirado no processo biológico de mutação genética.
- **Recombinação:** geram duas novas soluções por meio da recombinação de partes dos cromossomos de duas soluções iniciais. Simula o processo biológico de reprodução sexuada.

Depois de aplicados os operadores de evolução citados, acontece uma etapa de *seleção* dos indivíduos. Os indivíduos são selecionados por meio de uma *função de aptidão* (ou crítico) dependente de domínio, responsável por avaliar a qualidade da solução. Apenas os indivíduos considerados aptos (ou seja, as soluções de melhor qualidade) continuam no processo de evolução, sendo submetidos novamente aos

operadores de evolução acima. O processo continua até que seja atingido certo critério de parada, que pode ser baseado na quantidade de iterações ou qualidade das soluções da última geração.

### 2.1.2 Sistemas Rítmicos

Os algoritmos genéticos têm sido aplicados a um grande número de aplicações musicais na literatura (Todd & Werner, 1999). Essas aplicações consideram os trechos musicais como os *indivíduos* a serem evoluídos pelo algoritmo e a saída do sistema consiste no conjunto de indivíduos da geração final. Dentre esses sistemas, apenas quatro são voltados para a composição rítmica (Horowitz, 1994; Burton, 1998; Pazos et al., 1999; Pearce, 2000). Esses sistemas focam especificamente a criação de padrões rítmicos de bateria, a menos de (Pazos et al., 1999), que descreve um sistema inspirado nas performances puramente rítmicas polifônicas humanas.

Apesar dos quatro trabalhos usarem algoritmos genéticos, eles apresentam entre si algumas diferenças arquiteturais, como as estratégias de representação dos padrões rítmicos (cromossomos) e de escolha da população inicial. Porém, a maior e mais importante das diferenças entre os sistemas é a função de aptidão usada: Horowitz (1994) e Pazos et al. (1999) usaram avaliação humana, enquanto Burton (1998) e Pearce (2000) usaram redes neurais.<sup>2</sup>

Curiosamente, apenas os sistemas que usaram avaliação humana reportaram bons resultados. Horowitz (1994), avaliando subjetivamente seu próprio trabalho, afirma que os padrões gerados são comparáveis aos padrões de bateria distribuídos com a maioria das *drum machines* do mercado. Já Pazos et al. (1999) apresentam um experimento mais estruturado em que humanos com diferentes níveis de conhecimento musical classificaram a saída final do sistema. Os autores relataram que, após cerca de quatro gerações, a nota média atribuída ao sistema era 90, de um máximo de 100.

Por outro lado, Burton (1998) reporta uma grande homogeneidade (semelhança) entre os padrões gerados e Pearce (2000) afirma que a maioria dos padrões rítmicos gerados não se encaixava no gênero musical desejado (Drum'n'Bass), além de serem confusos e caóticos, em alguns casos.

---

<sup>2</sup> Horowitz (1994), na verdade, usou dois estágios de avaliação: humana, em um estágio inicial, e baseada em regras, em seguida. O importante, no entanto, é considerar o nível de interferência do usuário na avaliação (em contraste com a absoluta autonomia dos críticos baseados em redes neurais).

Assim, vemos que, apesar de a literatura relatar vários fracassos na aplicação de técnicas evolucionárias para outros sistemas de composição (conforme (Todd & Werner, 1999)), um relativo sucesso foi alcançado para os sistemas de composição puramente rítmica. Esse sucesso, no entanto, só foi possível graças ao uso de avaliação humana que, por outro lado, adiciona um problema conhecido na literatura como *fitness bottleneck* (Biles, 1994).

Esse termo se refere aos problemas que surgem quando humanos são usados para julgar a aptidão em um sistema musical baseado em algoritmos genéticos: o processo se torna lento, cansativo ao avaliador e pode gerar avaliações contraditórias. Mesmo sistemas com críticos automáticos também podem ser bastante lentos, como reportam Pearce (2000) e Todd & Werner (1999). Isso acontece devido ao próprio mecanismo dessa técnica, que requer várias rodadas de evolução e avaliação de um grande número de indivíduos, associado ao enorme tamanho do espaço de busca (o espaço de todas composições musicais possíveis, boas ou ruins).

Isso faz dos sistemas genéticos, em geral, inadequados para aplicações em tempo real. Na seção a seguir vamos mostrar uma outra abordagem para construção de sistemas musicais.

## **2.2 Sistemas Baseados em Agentes**

Iniciaremos esta seção apresentando uma introdução aos agentes inteligentes, na primeira subseção a seguir. A segunda subseção apresenta as abordagens para construção de sistemas musicais, enquanto a terceira subseção apresenta alguns *frameworks* de desenvolvimento de sistemas musicais multiagentes, encontrados na literatura. Por fim, a quarta subseção apresenta os sistemas de composição rítmica construídos sobre esse paradigma.

### **2.2.1 Introdução aos Agentes Inteligentes**

Não existe um consenso sobre a definição precisa de agentes inteligentes, mas Russell & Norvig (2003) dão a definição, talvez, mais básica do que são agentes inteligentes:

*“Tudo o que pode ser considerado capaz de perceber seu ambiente por meio de sensores e de agir sobre esse ambiente por intermédio de atuadores”* (pag. 33).

Outros autores, como Wooldridge (2002), acrescentam as características de autonomia de ação (ou de decisão) e de sociabilidade com outros agentes. De uma maneira geral, consideraremos os trabalhos que usam agentes no sentido da primeira definição, pelo menos. Porém procuramos dar ênfase aos trabalhos que possuem a característica adicional de sociabilidade. Chamaremos, de agora em diante, os sistemas formados de múltiplos agentes sociáveis de **sistemas multiagentes**.

Segundo Kelemen (2003), esse paradigma multiagente para construção de sistemas computacionais constitui-se numa promessa real de aumento do poder computacional em relação à computação algorítmica tradicional. Langen et al. (2004) sugere, ainda, que esse paradigma favorece a construção de sistemas computacionais criativos, por conta das características de autonomia e sociabilidade dos agentes que permitem que os agentes interajam entre si de diversas maneiras não previstas pelo projetista do sistema (ou do agente).

Diferente do que acontece com algoritmos genéticos, a aplicação do paradigma multiagente para composição não apresenta quaisquer restrições conceituais para sua aplicação no domínio musical. Questões relacionadas a tempo de execução, por exemplo, dependem de como o próprio agente é projetado. Por isso, estudaremos mais a fundo a aplicação dessa técnica ao domínio musical nas seções a seguir.

### **2.2.2 Modelagens de Sistemas Musicais**

O paradigma multiagente oferece diversas possibilidades de modelagem de sistemas musicais. Nesta subseção mostramos as principais abordagens encontradas na literatura para modelar um sistema musical baseado em múltiplos agentes. As abordagens aqui apresentadas se diferenciam pelo papel exercido por cada agente e pelo modelo de interação entre eles (organização).

Uma abordagem encontrada na literatura para modelar sistemas musicais multiagentes consiste em usar agentes para efetuar etapas distintas do processo de composição musical (Jewell et al., 2005; Fonseka, 2000). Nesses sistemas, cada agente encapsula um algoritmo com um diferente propósito musicológico dentro do processo de composição coletivo. Exemplos das diferentes etapas que podem ser realizadas pelos agentes são: definição do andamento, geração da harmonia e geração da melodia.

Conforme o caso, as etapas podem ser executadas paralelamente (independentemente) ou seqüencialmente (o resultado de uma etapa serve de entrada para outra etapa) e o resultado final do sistema é apenas aquele gerado pela última etapa.

Portanto, nessa primeira proposta de modelagem, os agentes precisam seguir uma ordem adequada na execução de suas tarefas de modo a gerar a composição final. Essa ordem pode ser elaborada pelos próprios agentes ou pode ser pré-definida pelo usuário.

Um detalhe ainda sobre essa primeira abordagem é que ela pode ser considerada a que mais se assemelha ao processo mental de criação, se considerarmos o modelo de cognição proposto por Minsky (analisado em (Singh, 2003)) que vê a mente como uma imensa sociedade de agentes bastante especializados que cooperam entre si na realização das atividades mentais humanas (e.g. composição musical).

A segunda abordagem consiste em atribuir a cada agente o papel de um músico em uma performance em grupo, como em (Masataka et al., 1996) e (Ramalho et al., 1997). Cada agente é, portanto, responsável pela geração de uma linha instrumental da peça final. Nessa abordagem, os agentes precisam trabalhar em sincronia (por meio de um metrônomo compartilhado ou por meio de técnicas de *beat tracking*) e seguir alguma referência musical mínima (geralmente a grade de acordes, tal como acontece em *jam sessions* reais). Trabalhos da área reportam a construção de agentes capazes de contribuir para uma performance musical fazendo desde bases rítmicas (Murray-Rust, 2003) até a improvisos de jazz (Biles, 1994). Notamos que, nessa abordagem, a interatividade com usuário humano é especialmente aplicável (e até desejável), uma vez que o papel de um agente é inspirado nos próprios músicos humanos.

Há ainda um conjunto de abordagens para sistemas compositores que busca inspiração em processos da natureza para modelar os agentes e suas interações (Todd & Miranda, 2003). Um desses sistemas, por exemplo, busca inspiração no processo de escolha de parceiros usado pelos pássaros (Todd & Werner, 1999). Tal sistema implementa agentes produtores de música (equivalente aos pássaros machos) e agentes ouvintes de música (fêmeas), responsáveis por julgar os produtores escolhendo aquele que mais lhe agrada para acasalar e produzir uma nova geração. Desse modo, os agentes evoluem musicalmente a cada geração.

Um outro exemplo de sistema, desenvolvido por Gueret et al. (2004), busca inspiração no mecanismo de formação de caminhos por meio de ferormônios, observado em colônias de formigas, para criar linhas melódicas. Cada vez que uma formiga passa numa certa aresta de um grafo, uma nota específica daquela aresta é tocada e a probabilidade da mesma aresta ser repetida futuramente por outra formiga é reforçada. Assim, à medida que uma formiga caminha, ela gera uma frase melódica que tende a ser repetida pelas formigas seguintes.

Esse grupo de abordagens inspiradas na natureza pode alcançar resultados imprevisíveis e variados, mas apresenta como grande desvantagem o fato de os agentes não terem funções claras do ponto de vista musicológico. Isso prejudica a qualidade musical dos resultados obtidos pela maioria desses sistemas.

Nesta subseção, apresentamos modelagens abstratas para construção de sistemas musicais, porém não entramos em detalhes infra-estruturais e de implementação de tais sistemas. Estes detalhes serão analisados na subseção a seguir.

### **2.2.3 Frameworks para Sistemas Musicais**

Sistemas multiagentes são sistemas relativamente complexos que requerem uma infra-estrutura de software que ofereça aos agentes, no mínimo, meios deles se localizam e se comunicarem. Adicionalmente, essa infra-estrutura pode oferecer outras características, como gerenciamento do ciclo de vida do agente, distribuição em rede (para permitir encontros entre agentes remotos) e mobilidade (o agente pode se transferir para outro nó da rede).

Essas infra-estruturas para sistemas multiagentes geralmente são encontradas na forma de *frameworks* de software independentes de aplicação. Na literatura, são encontrados vários desses *frameworks* para aplicações multiagentes de propósito geral, como Bellifemine et al. (1999) e Poslad et al. (2000), porém, nesta subseção, apresentamos alguns *frameworks* propostos especificamente para o domínio musical. Tais *frameworks* oferecem características adicionais necessárias às aplicações musicais, como: gerenciamento da saída sonora e desempenho adequado (para evitar atrasos na execução sonora).

O **Ligh Agent Framework** (Jewell et al., 2005), ou apenas **LAF**, é uma dessas propostas. O *framework* oferece uma plataforma distribuída para composição, onde cada agente encapsula um algoritmo de uma das etapas da composição (conforme abordagem citada na subseção anterior). Os agentes se comunicam entre si por certos canais síncronos (de entrada ou saída) chamados “portas”. Cada porta possui um nome específico, que indica o propósito dela (e.g. porta “andamento”), e transfere apenas dados bem tipados (e.g. boolean, int, string). Um agente pode ter várias portas de entrada e de saída e a organização dos agentes se dá pela interligação em cadeia de suas portas de entrada e saída. Essa ligação das portas acontece em tempo de execução, conforme a necessidade dos agentes. Os agentes localizam a porta de que necessitam pelo nome dela.

Fonseka (2000) propõe uma outra plataforma distribuída para agentes compositores em que cada agente encapsula um algoritmo de composição, na forma de um *script* Java. A comunicação entre agentes construídos sobre esse *framework* se dá apenas por meio de eventos MIDI *multicast* aos demais agentes. O autor usa as latências de rede entre os agentes como medida de “proximidade”, análoga à distância física entre músicos humanos. Inspirado nos corais, onde cantores fisicamente próximos servem de referência uns para os outros (enquanto cantores distantes não se influenciam), o autor usa essa latência para indicar a relevância das saídas sonoras entre cada par de agentes.

O **Andante** (Ueda et al., 2004) é a única plataforma encontrada que permite a migração dos agentes entre *hosts*. O autor argumenta que o uso de agentes móveis abre novas possibilidades para os sistemas musicais, como a criação de agentes musicais que se movem em um ambiente físico (um museu, por exemplo) acompanhando uma determinada pessoa.

O **VMMAS** (Virtual Musical MultiAgent System), desenvolvido por Wulforst e colegas (2003), é um modelo de desenvolvimento para sistemas multiagentes interativos em tempo-real. Segundo os autores, o objetivo do VMMAS é eliminar a separação conceitual entre humanos e computadores em sistemas musicais. Para garantir a uniformidade de interação, a comunicação se dá apenas por eventos MIDI, já que os humanos geralmente usam apenas um instrumento musical MIDI para fornecer entradas para sistemas musicais interativos. Para garantir a interação em tempo real, o sistema não oferece distribuição em rede e usa um *blackboard* como ponto central de comunicação, evitando *overheads* envolvidos em trocas de mensagens.

Na tabela 2.1, a seguir, apresentamos um resumo das características dos *frameworks* relacionados acima.

**Tabela 2.1:** Características dos *frameworks*

<i>Frameworks</i>	<i>Modelo de Comunicação</i>	<i>Linguagem de Comunicação</i>	<i>Distribuição</i>	<i>Mobilidade</i>
<b>LAF</b>	mensagens	tipos de dados: int, strings, etc.	Sim	Não
<b>Fonseka's</b>	mensagens	MIDI	Sim	Não
<b>Andante</b>	mensagens	específica de aplicação	Sim	Sim
<b>VMMAS</b>	blackboard	MIDI	Não	Não

Analisando a tabela 2.1 pode-se perceber que predomina o modelo de comunicação por meio de mensagens. Já o modelo *blackboard* aparece sob a justificativa de reduzir *overheads* de comunicação e uniformizar a troca de mensagens com humanos e, portanto, parece ser o modelo mais adequado para sistemas interativos. Quanto à linguagem de comunicação, não há um padrão universal, sendo o uso de mensagens MIDI a única abordagem que se repete. O problema dessa abordagem é que as mensagens MIDI são informações puramente musicais e, por isso, limitadas para indicar as intenções de um agente dentro de um processo de composição ou de uma performance em grupo. Apesar de haver trabalhos (e.g. Walker, 1997) em que agentes autônomos coordenam uma performance musical sem a troca de informações “extra-musicais” (i.e. não-MIDI), esse trabalho se limita a contextos musicais específicos (e.g. improvisação em jazz). Por outro lado, a troca de informações de níveis de abstração mais altos, como proposto por Murray-Rust & Smaill (2005), oferece aos agentes maior flexibilidade para ser adaptado a outros contextos.

Analisando a tabela 2.1 também é possível perceber que a característica de distribuição em rede é comum, de acordo com a idéia amplamente aceita de que uma das vantagens de usar uma abordagem multiagente é poder permitir a interação entre agentes heterogêneos desconhecidos através da internet (Murray-Rust, 2003; Kon & Iazzetta, 1998). No entanto, para haver interação entre agentes heterogêneos é necessária a adoção de padrões amplamente aceitos, o que não acontece nas plataformas apresentadas. Tanto a linguagem das mensagens como as próprias plataformas de execução são incompatíveis entre elas. Talvez por esse motivo, essas plataformas genéricas são projetos cuja aplicação se restringiu aos grupos de pesquisa que os desenvolveram.

Assim como acontece com os sistemas compositores de um modo geral, também são mais abundantes, dentro do paradigma multiagente, os sistemas musicais que focam harmonia e melodia. A subseção a seguir apresenta os dois únicos sistemas compositores baseados em agentes que focam na dimensão rítmica da música.

#### **2.2.4 Sistemas Rítmicos**

Nesta subseção, apresentamos os dois únicos sistemas rítmicos multiagentes encontrados na literatura: o VirtuaLatin (Murray-Rust, 2003) e o Rhythm Band Editor (Pachet, 2000).

O **VirtuaLatin** (Murray-Rust, 2003) é um sistema multiagente que simula uma banda de salsa, sendo cada agente um dos instrumentistas da banda. O foco do trabalho, contudo, foi o desenvolvimento do agente percussionista (ou “timbaleiro”) da banda, enquanto os demais agentes do sistema foram criados apenas para reproduzir trechos pré-gravados de cada instrumento. Além dos agentes musicistas, o sistema conta com um agente *maestro* responsável por controlar a saída de som do sistema, que é registrada em arquivo MIDI. Os agentes enviam, periodicamente, um compasso MIDI por vez para o agente *maestro*, que assume as tarefas de: redistribuir o compasso para os outros agentes, mesclar os compassos recebidos de cada agente e enviar o compasso mesclado para o arquivo MIDI de saída.

O agente timbaleiro é configurado para receber como entrada (por meio do agente *maestro*) todos os compassos produzidos pelos demais agentes. Usando essa informação puramente musical, o agente aplica um algoritmo para gerar um arranjo de percussão de salsa adequado ao contexto musical em questão, para enviar ao *maestro*, em seguida. Baseado no GTTM (Lerdahl & Jackendoff, 1983), esse algoritmo é capaz de extrair várias informações de alto nível da peça por meio de várias etapas de análise (de atividade, de harmonia, rítmica e de paralelismo), executadas a cada compasso. Partindo dessa representação de alto nível, o sistema aplica um conjunto de regras para selecionar, de um conjunto de padrões rítmicos típicos de salsa, aquele que melhor se adequa ao compasso.

Para avaliar o sistema, o autor realizou experimentos com o público em geral e com músicos de salsa. Nesses experimentos, uma música foi apresentada em duas versões: uma com as percussões tocadas pelo VirtuaLatin e outra tocada por um humano. O resultado mostrou que o VirtuaLatin consegue enganar o público em geral, mas não consegue enganar músicos experientes. Como aplicação para usuário final, o VirtuaLatin pode servir, portanto, como uma ferramenta para geração de arranjo rítmico de salsa de qualidade razoável. O uso desse sistema, no entanto, fica limitado apenas aos músicos de salsa, uma vez que o sistema foi construído usando intenso conhecimento específico de salsa.

Quanto ao processo de criação, o VirtuaLatin ainda usa um algoritmo centralizado (no agente timbaleiro) para gerar o arranjo rítmico. Portanto, o autor não aproveita as vantagens da abordagem multiagente de criar resultados diversos por meio de agentes que interagem de maneira flexível.

O outro sistema rítmico multiagente encontrado é o **Rhythm Band Editor** (Pachet, 2000), cujo objetivo é o de estudar ritmos enquanto estruturas evolutivas. O sistema permite estudar regras para criação ou evolução de polifonias rítmicas. Cada agente do sistema é inicialmente configurado com as seguintes informações:

- **Notas MIDI** (de algum instrumento de percussão) que ele pode tocar. Por exemplo, um agente pode ser configurado para tocar agogô se forem adicionadas as notas<sup>3</sup>: G5 (nota aguda de agogô) e Ab5 (nota grave).
- **Um padrão rítmico inicial**, que pode ser vazio ou formado por sucessões de instâncias das notas cadastradas.
- **Um conjunto de regras de transformação**, aplicáveis aos padrões rítmicos. Cada regra possui uma pré-condição que, quando satisfeita, dispara uma ação de edição do padrão rítmico do agente.

Para permitir interações entre os agentes, o sistema mantém uma partitura global (comunicação *blackboard*) com as performances musicais dos agentes em formato MIDI. Um *módulo de percepção* presente nos agentes recebe como entrada a partitura global e gera um conjunto de informações de alto nível, tais como: estrutura de batidas e batidas acentuadas. Um outro módulo, chamado de *módulo de produção*, utiliza a base de regras fornecida na criação do agente para alterar o padrão rítmico tocado por ele.

As regras de transformação têm suas pré-condições especificadas a partir das informações de alto-nível geradas pelo módulo de percepção e, quando disparadas (pré-condições satisfeitas), executam alguma ação de edição (adição ou remoção de notas) no padrão tocado pelo agente. Um exemplo de regra é: “*se existe uma batida forte na qual o agente não toca, então adicione uma nota nessa batida*”.

Depois que o usuário do Rhythm Band Editor especifica, para cada agente, as notas possíveis, o padrão inicial e o conjunto de regras de transformação, o sistema pode ser iniciado. Os agentes entram, então, em um ciclo que alterna a execução de seus respectivos padrões rítmicos com a aplicação das regras de transformação para evoluir os padrões para o ciclo seguinte. O sistema permanece nesse ciclo até que o sistema convirja para uma situação em que nenhuma regra é aplicável. Assim, o sistema permite aos usuários testar diferentes regras e ouvir imediatamente os efeitos dessas regras.

Vemos, portanto, que o objetivo do Rhythm Band Editor não é a geração de saídas musicais criativas, mas o estudo das regras e seus efeitos sobre uma performance

---

<sup>3</sup> Assumindo notas no canal 10, no padrão General MIDI.

rítmica polifônica (e.g. quais regras devem ser aplicadas a cada componente da bateria para gerar um ritmo característico de *rock*). A usabilidade desse sistema também é relegada a segundo plano, como pode ser visto pelo intenso conhecimento musicológico necessário para controlar os resultados gerados pela aplicação.

Por outro lado, o sistema é pioneiro em pensar que cada instrumento de percussão pode estar associado a regras diferentes. Sistemas rítmicos anteriores tentavam julgar todo o resultado de maneira centralizada, sem levar em consideração diferenças dependentes de instrumento ou de gêneros musicais, por exemplo.

### **2.3 Considerações Finais**

Analisamos, neste capítulo, duas classes de sistemas compositores rítmicos encontrados na literatura: os sistemas baseados em algoritmos genéticos e os sistemas multiagentes. Dos sistemas baseados em algoritmos genéticos, apenas aqueles baseados em avaliação humana reportaram bons resultados. Porém, esse processo de avaliação humana é cansativo ao avaliador e insere um gargalo de desempenho no sistema.

O paradigma multiagente, por outro lado, oferece uma grande flexibilidade de abordagens para o domínio musical. Mostramos as modelagens típicas e analisamos as plataformas genéricas propostas na literatura para construção de sistemas musicais. Por fim, analisamos os dois únicos sistemas encontrados na literatura que usam a abordagem multiagente para gerar composições rítmicas: o VirtuaLatin e o Rhythm Band Editor.

O VirtuaLatin implementa um algoritmo de acompanhamento rítmico de maneira centralizada em um único agente e, portanto, não faz uso das possíveis vantagens dos sistemas multiagentes. Além disso, o sistema admite um único estilo apenas (a salsa). Já o Rhythm Band Editor é baseado num mecanismo de interação indireta entre os agentes, no qual os agentes alteram uma mesma partitura global. Este sistema é flexível para permitir gerar resultados diversificados, mas, para isso, exige um conhecimento muito grande de regras e seus efeitos.

## CAPÍTULO 3

# Representação e Avaliação Rítmicas

No capítulo anterior, apresentamos uma revisão dos sistemas musicais inteligentes, com especial destaque para os sistemas rítmicos. Neste capítulo, discutiremos mais a fundo as técnicas de representação e avaliação de padrões rítmicos, fundamentais na construção destes sistemas. Com vistas a alcançar resultados interessantes para a percepção musical humana, adotamos, em todo o estudo aqui apresentado, a música não como uma objeto de primeira classe pronto para ser diretamente analisado, mas como um fenômeno cognitivo complexo induzido na mente humana pela apresentação de certos sons. Portanto, sempre que possível, privilegiamos nesse capítulo os trabalhos fundamentados em considerações sobre a cognição musical.

Na seção 3.1, apresentamos um estudo sobre a estrutura presente na dimensão rítmica da música, que servirá de referencial para as seções seguintes. Em seguida, na seção 3.2, abordamos as representações rítmicas mais relevantes encontradas. Por fim, na terceira seção deste capítulo, falamos de métodos de avaliação de padrões rítmicos.

### **3.1 Estrutura Rítmica**

Apesar de haver, na literatura, um grande número de trabalhos sobre a percepção rítmica, ainda não foi desenvolvida uma teoria amplamente aceita sobre essa dimensão musical, conforme Weyde (2001) e Honning (2002). Em contraste, as dimensões melódica e harmônica da música são relativamente bem estudadas por meio de conceitos como: modos, escalas, acordes e análise harmônica funcional de acordes (Gómez et al., 2005). Outros trabalhos também expressam opinião semelhante, como (Pachet, 2000), (Pfleiderer, 2003) e (Toussaint, 2003a).

Para tentar entender melhor a dimensão rítmica da música, analisamos alguns trabalhos que tentam discriminar os elementos que formam a estrutura rítmica da música,

ainda que (provavelmente) esses trabalhos não ofereçam mais do que uma ponta de iceberg de toda a verdade por trás desse assunto.

### 3.1.1 Elementos Básicos

Um elemento estrutural dos padrões rítmicos que apresenta grande aceitação entre os pesquisadores é uma série de **pulsos isocrônicos** subjacentes em uma audição musical (Parncutt, 1994). Esse pulso serve como uma grade temporal, ou um *clock*, que demarca os instantes mais prováveis para ocorrência de ataques de notas. Essa grade é induzida pela mente humana sempre que ouvimos algum evento temporal minimamente regular. O pulso isocrônico é o que nos leva a bater regularmente com o pé ou com a mão quando ouvimos uma música, por exemplo.

Pressing (2002) apresenta uma justificativa evolutiva para a indução dessa grade isocrônica: ela teria surgido como método de previsão de eventos temporais que serviria, por exemplo, para permitir aos seres humanos primitivos ouvir a aproximação de um animal e atacá-lo de surpresa. Uma vez que a resolução temporal da audição humana é bastante superior à da visão, aquele teria sido o sentido ideal para o surgimento desse modelo de previsão de eventos temporais.

Eventos temporais, como os ataques de notas em música, ocorrem na dimensão contínua do tempo, obviamente. Porém, a indução da grade de pulsos isocrônicos associa os eventos contínuos aos pulsos, que são discretizações do tempo. Isso dá aos padrões rítmicos uma natureza dual: discreta (relacionado ao pulso em que ocorre o evento) e contínua (o momento exato do ataque, dentro do pulso). Honing (1993) afirma que, de maneira geral, os elementos discretos da música carregam estrutura, enquanto os elementos contínuos carregam expressividade<sup>4</sup>. A natureza discreta dos padrões rítmicos permite a sua representação em modelos simbólicos, como a notação musical ocidental. Porém, na hora de executar uma peça musical representada nessa notação, um (bom) músico humano fará pequenos desvios temporais para tornar a performance mais atraente. Honing (2002) afirma que um ouvinte humano é capaz de distinguir essas duas informações: a **categoria rítmica** discreta e os **desvios temporais** contínuos dentro do pulso.

---

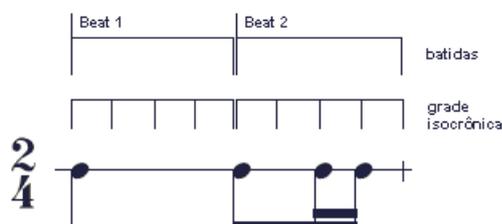
<sup>4</sup> Isso também acontece com as frequências das notas que, apesar de precisamente calculadas segundo um diapasão, em performances reais costumam sofrer micro-variações (se o instrumento permitir) para dar maior expressividade à interpretação.

É comum uma execução musical apresentar variações de velocidade no decorrer de uma performance. Por exemplo, o *ritardando* é uma técnica usada dar uma sensação de finalização ou resolução, especialmente no final de uma música, e que consiste em diminuir gradualmente a velocidade da execução. Isso traz à luz outro elemento temporal importante no estudo dos ritmos (e da música em geral): o **andamento**, que pode ser definido como a impressão de velocidade da música, ou a taxa de pulsos por unidade de tempo. Alguns pesquisadores consideram os desvios temporais contínuos como variações do andamento. Novamente, Honing (2002) apresenta argumentos de que são dois elementos distintos para a percepção humana.

Portanto, concluímos que são quatro os elementos básicos da percepção rítmica: a grade de pulsos isocrônicos, a categoria rítmica (discreta), os desvios temporais expressivos (contínuos) e o andamento.

### 3.1.2 Métrica

Na verdade, diversos níveis de pulsos regulares podem ser induzidos pela audição de uma mesma música. Aos pulsos do nível de menor granularidade que podem ser induzidos chamaremos de pulsos atômicos ou tatums (Iyer et al., 1997). Os pulsos atômicos costumam assumir sucessivas importâncias relativas distintas (e.g. probabilidade de ocorrência ou de acentuação dos ataques) que se repetem regularmente. Isso induz na percepção humana ao agrupamento dos pulsos atômicos em pulsos maiores ou batidas (figura 3.1). Essa seqüência de importâncias relativas se repete ciclicamente após um certo número de pulsos atômicos, caracterizando o que convencionou-se chamar de compasso. Assumimos que os pulsos maiores, suas sub-divisões em pulsos atômicos e as importâncias relativas entre os pulsos caracterizam o conceito de métrica.



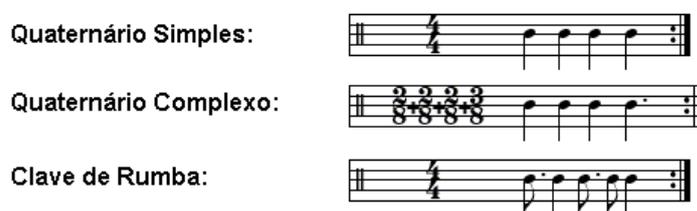
**Figura 3.1:** Batidas (pulsos maiores) e grade isocrônica (pulsos atômicos)

Na teoria musical tradicional, os pulsos maiores da métrica são chamados “batidas” e, geralmente, são representados como semínimas ou colcheias. Pulsos

atômicos típicos são colcheias e semi-colcheias. As métricas mais comuns na música ocidental são as **métricas regulares** formadas por duas (binária), três (ternária) ou quatro batidas (quaternária) de igual tamanho. Porém, em algumas manifestações musicais modernas ou étnicas, aparecem as chamadas **métricas irregulares ou complexas**, que são métricas formadas por pulsos maiores com diferentes tamanhos, em número de pulsos atômicos (e.g. London, 2000). Este mesmo trabalho define que as métricas complexas devem apresentar a característica adicional de serem **conjuntos maximamente espaçados**, ou seja, os pulsos maiores, apesar de desiguais, devem ser posicionados o mais distante possível uns dos outros.

Há ainda um conjunto de padrões rítmicos, chamados de **ritmos clave** ou apenas **claves**, também formados por ataques espaçados irregularmente e que apresentam função semelhante às métricas complexas (a ser discutida adiante). Os ritmos clave são executados em um instrumento também chamado de claves (blocos de madeira), ou em *cowbells*, e ocorrem principalmente em manifestações musicais africanas e caribenhas. No entanto, eles não podem ser considerados pulsos de uma métrica, porque seus ataques não são maximamente espaçados. Pressing (2002) argumenta ainda que esses ritmos dão a impressão de serem fundamentados sobre uma métrica regular. Outra diferença em relação às métricas simples ou complexas é o fato das claves serem padrões rítmicos efetivamente executados, enquanto as métricas são fenômenos induzidos (apesar de poderem ser executados por algum instrumento).

A figura 3.2 abaixo apresenta um exemplo de cada um dos três “esquemas de pulsos” apresentados:



**Figura 3.2:** Três diferentes esquemas de pulsos: uma métrica regular, uma métrica complexa e um ritmo clave caribenho

Segundo London (2004), a métrica envolve tanto nossa percepção inicial das batidas como também uma preparação para a ocorrência das batidas no futuro (antecipação). Na verdade, isso acontece não apenas com as métricas simples, como

também com as métricas complexas (London, 2000) e com os ritmos claves (Toussaint, 2002). Isso faz com que esses três fenômenos citados sirvam como *frameworks* de referência para a organização temporal da performance musical. Em outras palavras, esses três diferentes fenômenos funcionam como metrônomos que guiam todas as linhas rítmicas dos instrumentos envolvidos em uma performance musical.

Ressaltamos que esse tipo de *framework* já se situa em um nível de abstração acima da grade isocrônica de pulsos atômicos considerada inicialmente (vide figura 3.1). Na verdade, a partir de uma mesma grade isocrônica podem ser induzidas várias métricas, a depender de como os diferentes pulsos atômicos são enfatizados. Em algumas situações, diferentes métricas paralelas podem ser induzidas de diferentes instrumentos de uma mesma performance, o que caracteriza o conceito de **polirritmos**. Na figura 3.3, mostramos uma performance polirrítmica com um instrumento tocando uma métrica de quatro pulsos e outro tocando uma métrica de seis.



**Figura 3.3:** Polirritmo 4:6 (quatro contra seis)

Com respeito às durações dos pulsos métricos, London (2000) compilou da literatura um conjunto de restrições identificadas por meio de experimentos com humanos:

- a. Pulsos atômicos devem ter duração mínima de 100 ms.
- b. A duração total dos pulsos da métrica não deve ultrapassar 5s.
- c. De preferência, pulsos maiores devem se situar na faixa de 500 a 900 ms.

Essas restrições têm implicações nas alterações de andamento que são possíveis para cada métrica, pois a mudança de andamento pode tirar algum pulso dos “limites”, alterando completamente a percepção dos ritmos. Devido à presença de pulsos de diferentes durações, London afirma que as métricas complexas podem ser mais sensíveis ao andamento.

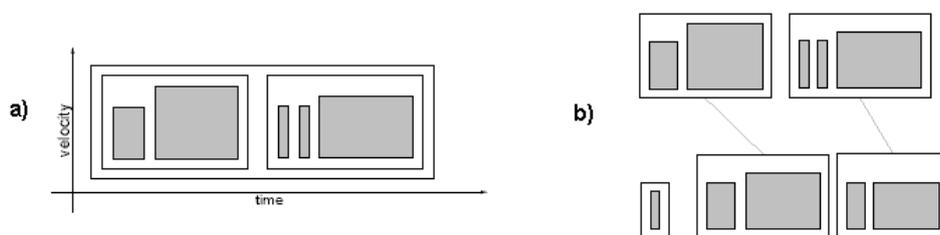
### 3.1.3 Relações Estruturais

A percepção musical como um todo é um fenômeno ainda pouco compreendido. Sabe-se, no entanto, que, para a mente humana, a música carrega uma complexa estrutura

que vai além daquilo que é expresso na notação tradicional. Alguns estudos, por exemplo, revelam certas relações geométricas e matemáticas em trabalhos de compositores eruditos, especialmente Bach (Fucks, 1962; Hofstadter, 1979; Hsu & Hsu, 1990). A falta de preocupação com essa estrutura musical é uma das causas prováveis do fracasso de muitos trabalhos de composição algorítmica.

Uma parte significativa da estrutura musical certamente está relacionada com a sua dimensão rítmica. Já vimos, nas subseções anteriores, alguns dos elementos estruturais mais bem aceitos na literatura, porém existem pesquisas que buscam tanto identificar outros elementos estruturais, como também associar os elementos estruturais entre si, identificando relações entre os elementos. A abordagem mais comum para relacionar os elementos estruturais da música costuma envolver **agrupamentos hierárquicos**. Weyde (2001) e Lartillot (2003) afirmam que notas podem ser agrupadas em “motivos”, que por sua vez podem ser agrupados em “frases” (vide figura 3.4). Os motivos podem ainda ser relacionados entre si por relações de repetição, seqüência, variação ou contraste.

O trabalho de Bertinotti (2004) apresenta uma fundamentação teórica para a existência desses agrupamentos. Porém, não há um entendimento claro de como a mente humana efetivamente realiza esses agrupamentos, nem de como cria as relações entre eles.



**Figura 3.4:** Agrupamentos hierárquicos (em motivos e frases) e relações entre eles

Pressing (2002) fala de relações estruturais presentes em ritmos africanos e derivados, que seriam análogas a fenômenos do discurso humano. Uma dessas relações é a de **chamada-e-resposta** em que uma frase curta é apresentada (a chamada), sendo imediatamente seguida por outra frase idêntica ou complementar (a resposta). Chamada e resposta podem ser executadas por um mesmo instrumento ou por instrumentos distintos. Alguns trabalhos consideram os ritmos clave como exemplos de relações de chamada-e-resposta. No ritmo clave de rumba mostrado na figura 3.2, por exemplo, os dois primeiros ataques seriam a “chamada” e os três ataques seguintes seriam a “resposta”.

## 3.2 Representações Musicais

Várias áreas de pesquisa, como a Computação Musical, a Musicologia e a Cognição Musical (Psicologia), têm interesse direto na especificação de um modelo adequado de representação da música em geral (Honning, 1993). O modelo escolhido tem um grande impacto na maneira como os trechos musicais podem ser manipulados e, por conseguinte, na viabilidade da construção de sistemas computacionais musicais.

Nas seções a seguir, apresentamos diferentes técnicas para representar os diversos elementos temporais da música apresentados na seção anterior. A primeira subseção apresenta alguns modelos de representação dos padrões rítmicos. Em seguida falamos de representações da métrica. Por fim, discutiremos a identificação e a representação de relações estruturais. A ordem adotada é análoga à ordem em que esses diversos conceitos foram apresentados na seção anterior.

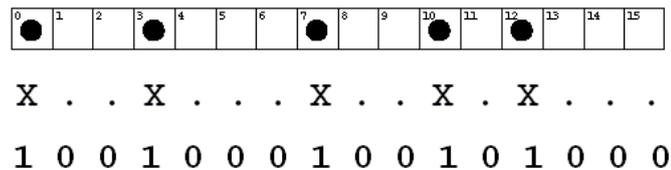
### 3.2.1 Representação dos Padrões Rítmicos

Como ponto de partida, vamos considerar a notação de padrões rítmicos na notação tradicional (partituras). Nesta notação, existem diferentes maneiras de se representar um mesmo padrão rítmico, pois basta posicionar corretamente os ataques do padrão usando qualquer figura de nota (Toussaint, 2002). Mostramos, na figura 3.5, dois exemplos de notação tradicional para o ritmo clave de *Rumba*. O segundo exemplo toma o pulso isocrônico induzido (uma semínima, no caso) como figura de nota padrão.



Figura 3.5: Notações alternativas para o ritmo-clave de Rumba

Partindo desta última notação, quantizada em semínimas, é possível derivar uma notação chamada de **TUBS** – Time Unit Box System (Toussaint, 2002), que pode ser vista na figura 3.6 representando o mesmo padrão rítmico da figura anterior.



**Figura 3.6:** Notação TUBS e variantes

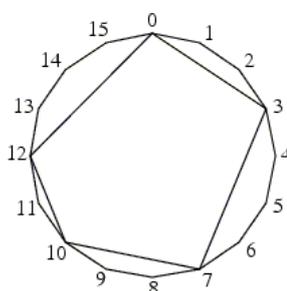
A representação TUBS pode ser entendida como um vetor em que cada posição tem um símbolo de ataque ou um símbolo de pausa (que aqui não denota necessariamente o silêncio, mas simplesmente um período entre ataques). Por exemplo, a última representação da figura 3.6 usa ‘1’ para indicar ataque e ‘0’ para pausa em cada posição do vetor TUBS. As principais vantagens dessa notação são: a representação explícita dos pulsos isocrônicos e a simplicidade, que permite que ela seja utilizada até mesmo por músicos que não sabem partituras (Toussaint, 2004a).

A partir da representação TUBS é possível extrair várias outras representações vetoriais. Uma delas é o **vetor de posições de ataque**, que pode ser obtido colocando, em cada posição  $i$  do vetor, o momento (ou seja, a posição na representação TUBS) em que ocorre o  $i$ -ésimo ataque. O ritmo clave de *Rumba*, por exemplo, seria representado como  $\langle 0,3,7,10,12 \rangle$ .

Porém, é bastante provável que os intervalos entre ataques tenham maior relevância para a percepção humana do que as posições de cada ataque. Assim, outra representação vetorial que pode ser derivada é o **vetor de intervalos entre ataques** (Toussaint, 2002), que mantém, em cada posição, o número de “caixas” existentes entre ataques consecutivos. Essa representação pode ser obtida a partir do vetor de posições ataques simplesmente subtraindo cada valor do valor seguinte. O ritmo clave de *Rumba* nessa representação ficaria assim: [3,4,3,2,4].

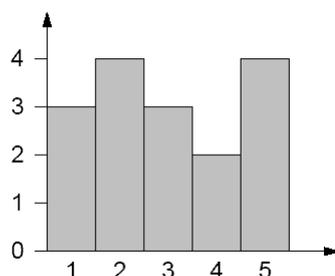
Uma outra abordagem para representação rítmica consiste em utilizar elementos geométricos para dar uma percepção visual (minimamente) análoga à percepção sonora musical. Um exemplo de representação geométrica é a **representação poligonal** (Toussaint, 2002) que representa um padrão como um polígono inscrito em uma circunferência, baseando-se no fato de os padrões rítmicos de percussão serem executados ciclicamente. Nessa representação, uma circunferência é subdividida em pontos equidistantes que representam os pulsos isocrônicos. Contando a partir do ponto mais alto da circunferência, os pontos de ataques são marcados (as pausas são os pontos

ignorados). O polígono que representa o padrão é obtido pela ligação dos pontos marcados. Na figura 3.7, mostramos o ritmo clave de *Rumba* nessa representação.



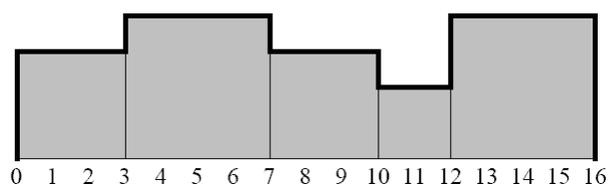
**Figura 3.7:** Representação poligonal do ritmo clave de Rumba

Outras representações geométricas encontradas são baseadas em histogramas. Aquela que é, talvez, a mais simples dessas abordagens é exemplificada na na figura 3.8 e consiste em representar como barras do histograma cada posição do vetor de intervalos entre ataques mostrado anteriormente.



**Figura 3.8:** Histograma de intervalos entre ataques do ritmo-clave de Rumba

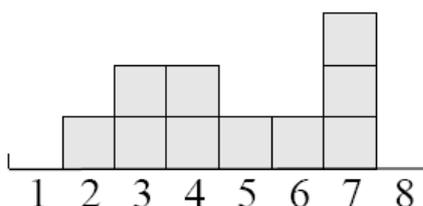
Outra representação baseada em histogramas, chamada **TEDAS** (Temporal Elements Displayed As Squares), foi proposta no estudo do “ritmo da fala” (Gustafson, 1988) e usa os dois eixos para representar o tempo (Toussaint, 2002). No eixo vertical, semelhantemente à abordagem anterior, são representados os intervalos entre ataques sucessivos. No eixo horizontal os pontos onde ocorrem os ataques delimitam as barras do “histograma”, deixando-as quadradas. Assim, a representação TEDAS tem como vantagens: manter a precisão temporal no eixo horizontal e permitir uma visualização relativa dos intervalos entre ataques, no eixo vertical. Mostramos na figura 3.9, a representação TEDAS do mesmo padrão rítmico considerado antes.



**Figura 3.9:** Ritmo clave de Rumba na representação TEDAS

Para facilitar a representação computacional do gráfico TEDAS, pode-se usar o chamado **vetor cronotônico**, que guarda, para cada pulso atômico, o valor (dimensão vertical) do gráfico **TEDAS** (Toussaint, 2004a). O ritmo clave de Rumba representado como vetor cronotônico ficaria assim: {3,3,3,4,4,4,4,3,3,3,2,2,4,4,4,4}.

Ainda outra representação baseada em histograma é o **espectro de freqüências de distâncias entre ataques**, também chamado de histograma completo de intervalos. Ele é calculado a partir das distâncias entre todos os pares de ataques, considerando a possibilidade de rotação (i.e. contar até o último ataque e retornar ao primeiro). Na figura 3.10, mostramos a clave Rumba nessa representação.



**Figura 3.10:** Histograma de freqüências das distâncias entre ataques

Cada uma das representações apresentadas anteriormente destaca diferentes aspectos da música e pode ser mais adequada para descobrir diferentes propriedades estruturais da música (como veremos na seção 3.3). Portanto, o desenvolvedor de um sistema computacional deve considerar a possibilidade de escolher não apenas uma dessas representações, mas várias representações paralelas, de modo que cada representação alternativa descreva independentemente o mesmo padrão rítmico (Honing, 1993). O uso de representação múltipla permite unir as vantagens de cada modelo.

### 3.2.2 Representação do Andamento

A notação tradicional representa o andamento basicamente por meio de adjetivos em italianos, tais como: *largo* (muito lento), *moderato* (moderado) e *presto* (muito rápido). Além disso, mudanças de andamento também são representadas por nomes como: *rallentando* (execução gradativamente se torna mais lenta) e *accelerando*

(execução gradativamente mais rápida). Originalmente, essa representação apresentava como principal vantagem a intuitividade na leitura dos valores do andamento, assumindo músicos que entendem italiano. Por exemplo: é mais fácil entender o que é um andamento “moderado” do que um andamento de 120 BPM (semínimas por minuto). Além disso, essa representação oferece alguma liberdade para intérpretes humanos, por não definir precisamente a velocidade real da execução, como mostramos na tabela 3.1.

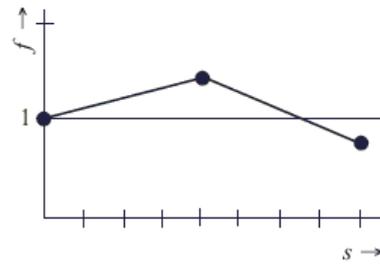
<i>Andamento em Italiano</i>	<i>Andamento Real</i>
Largo	40 – 54 bpm
Adagio	66 – 76 bpm
Andante	76 – 92 bpm
Moderato	108 – 120 bpm
Allegro	120 – 168 bpm
Presto	168 – 208 bpm

**Tabela 3.1:** Andamento real para algumas indicações em italiano

Por outro lado, a imprecisão dessas indicações em italiano traz dificuldades para a criação de sistemas computacionais. Por exemplo: como decidir o andamento real preciso para um dado adjetivo “*adagio*” e como classificar em adjetivo um dado andamento real? E como devem acontecer as mudanças de andamento (*rallentando* e *accelerando*)? O sistema tradicional não define especificações precisas para isso, havendo até mesmo sobreposição entre as faixas de andamentos reais executadas por diferentes humanos para alguns dos adjetivos de andamento.

Não existem muitas alternativas para representação do andamento. A literatura propõe que a mente humana percebe tanto a velocidade da execução (o andamento) como suas variações no tempo e, com base nisso, propõe uma **curva de andamento**, que é uma representação funcional do andamento. Essa função teria como domínio a posição temporal no modelo da representação (e.g. o compasso e a posição dentro do compasso) e como imagem o valor real do andamento naquele ponto da execução. É importante salientar, no entanto, que a curva só pode mudar de direção a cada pulso atômico, pois mudanças de velocidade intra-pulsos são percebidas como desvios expressivos (Honing, 2002), conforme apresentamos na seção anterior. A vantagem dessa representação é que ela define precisamente os valores de andamento e ainda permite a visualização de como

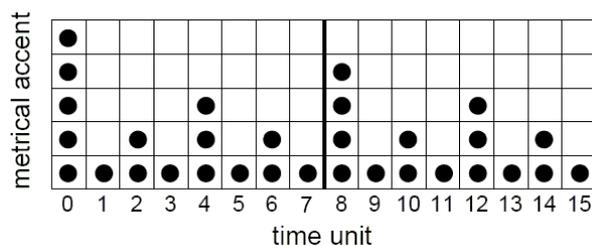
o andamento evolui ao longo da performance. Mostramos na figura 3.11 um exemplo de curva de andamento.



**Figura 3.11:** Curva de andamento. O eixo  $s$  representa a posição temporal na partitura (em pulsos atômicos) enquanto o eixo  $f$  representa um fator multiplicativo do andamento real.

### 3.2.3 Representação da Métrica

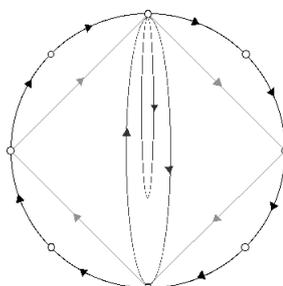
Na notação tradicional, a métrica é representada como uma fração. Para as métricas simples, por exemplo, a figura de nota que representa um pulso maior (batida) é usada no denominador da fração, enquanto o numerador indica o número de pulsos por compasso. London (2000) argumenta que essa é uma representação indireta da métrica. Ela falha por não representar as subdivisões da batida (pulsos menores) e as importâncias relativas entre batidas (ou entre pulsos menores) de maneira explícita. Por esse motivo, Lerdahl & Jackendoff (1983) propõem a **estrutura métrica**, que é um vetor de pesos que representa explicitamente as importâncias relativas entre os pulsos, como pode ser visto na figura 3.12.



**Figura 3.12:** Estrutura métrica para uma métrica quaternária regular

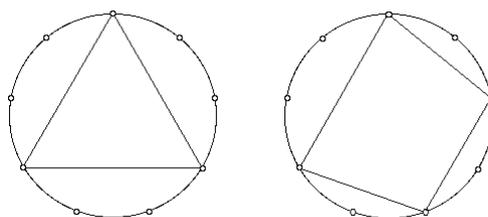
London (2000), apesar de reconhecer a importância da estrutura métrica de Lerdahl & Jackendoff, observa que falta ao modelo deles a representação de um aspecto crucial da métrica – a ciclicidade. Com esse argumento, ele propõe uma **representação cíclica da métrica** que guarda algumas semelhanças com a representação poligonal dos padrões rítmicos apresentada previamente. De modo semelhante, ambas as representações são baseadas em uma circunferência dividida em arcos de mesmo

tamanho, que representam os pulsos atômicos isocrônicos. Dentro dessa circunferência um polígono é inscrito pela ligação dos pontos onde ocorrem as batidas. Porém, na representação cíclica da métrica, outras figuras (curvas fechadas) são inscritas recursivamente para representar níveis superiores sucessivos da métrica.



**Figura 3.13:** Representação cíclica de um métrica quaternária regular

Na figura 3.13, mostramos uma métrica quaternária simples baseada em uma grade isocrônica de oito pulsos atômicos. O quadrado inscrito na circunferência representa o nível de batidas (pulsos maiores), enquanto as duas curvas interiores representam, respectivamente, os níveis de meio-compasso e de compasso. Na figura 3.14, mostramos, usando a representação cíclica, duas métricas baseadas em uma mesma grade isocrônica de nove pulsos. A primeira figura representa uma métrica ternária, enquanto a segunda representa uma métrica complexa quaternária.



**Figura 3.14:** Duas métricas possíveis em uma grade de nove pulsos

London (2000) ressalta que a vantagem de sua representação é uma percepção visual mais precisa. Por outro lado, a desvantagem em relação à estrutura métrica de Lerdahl & Jackendoff (1983) é a omissão dos pesos relativos dos pulsos maiores.

### 3.2.4 Representação das Relações Estruturais

Conforme apresentado na seção anterior, a estrutura musical é complexa e envolve não apenas certos elementos estruturais, como também certas relações entre

esses elementos. Vimos que, para os elementos estruturais mais bem aceitos na literatura, a literatura reporta diversas alternativas de representação. Esse não é o caso das relações estruturais, que ainda não foram identificadas pelas pesquisas da percepção musical humana. Por isso, elas não são diretamente representadas na notação tradicional nem na maioria dos modelos de representação pesquisados. (Honing, 1993) reconhece essa carência nos modelos e faz uma discussão sobre possíveis estratégias para representação de estrutura musical. Contudo, ele não vai além de uma discussão abstrata de como inserir essa característica nos modelos de representação musical.

### ***3.3 Extração de Propriedades Rítmicas***

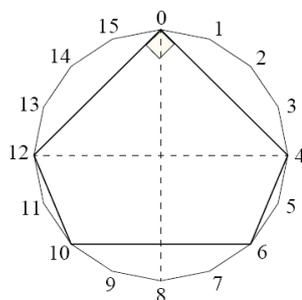
Como comentamos no final da seção 3.1, a identificação de relações estruturais em padrões rítmicos é importante para uma melhor compreensão deste fenômeno. Apesar da dificuldade em definir e indentificar as relações estruturais relevantes, a literatura identifica certas características dos padrões rítmicos, que possivelmente revelam algo da estrutura subjacente. Tipicamente, tais características são expressas por valores numéricos e obtidas por meio de algum método de cálculo, aqui chamados de “medidas”. Elas podem servir como uma maneira indireta de medir a qualidade de um padrão rítmico em determinados contextos, como veremos ao analisar cada medida.

As subseções a seguir tratam de cada uma das propriedades rítmicas encontradas na literatura, mostrando as diversas medidas existentes para calculá-las. Nos casos onde existem várias medidas, procuramos selecionar apenas as mais relevantes com base em experimentos relatados na literatura. Cada uma das medidas é baseada em algum dos modelos de representação apresentados na seção anterior.

#### **3.3.1 Propriedades Geométricas**

As representações geométricas apresentadas na seção anterior abrem possibilidades de aplicação de um grande leque de ferramentas matemáticas para extração de características dos padrões rítmicos. Toussaint (2002) sugere que as propriedades geométricas de notações rítmicas podem ser úteis para comparação, análise e classificação automática de ritmos. Nesse mesmo trabalho, usando a representação poligonal dos ritmos, Toussaint identificou três propriedades geométricas capazes de identificar unicamente cada padrão rítmico de um conjunto de seis padrões africanos, caribenhos e brasileiros analisados. As características são: presença de ângulo reto interno, número de triângulos isóceles e presença de eixos de simetria. Em especial, a

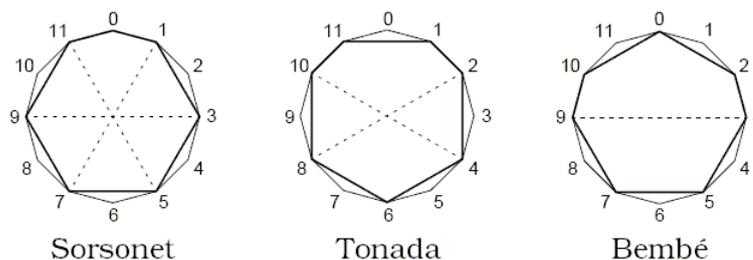
presença do ângulo reto interno foi capaz de separar os padrões africanos dos demais, o que pode sugerir que essa é uma medida de preferência de ritmos dentro da cultura africana. Mostramos na figura a seguir um padrão rítmico com as três características citadas. As retas tracejadas indicam a base de um triângulo isóceles (eixo 12-4) ou um eixo de simetria (eixo 0-8).



**Figura 3.15:** Padrão rítmico africano Shiko (Toussaint, 2002)

Toussaint (2004b) propõe ainda a análise de histogramas de frequências de distâncias entre ataques de um padrão rítmico para identificar certas qualidades perceptuais dos padrões. Ele sugere, por exemplo, que (pag. 6): “*rhythms with relatively uniform (flat) histograms, and few, if any, gaps may be interesting from the musicological point of view*”. O problema dessa análise bem com das outras análises geométricas apresentadas, entretanto, é justamente a falta de interpretações musicológicas claras. O que Toussaint chama de “*interesting*” pode não ser interessante para outro ouvinte.

Uma outra propriedade geométrica que pode ser mais útil é o que Toussaint (2003b) chama de “colar” (*necklace*), que são classes de equivalência de ritmos considerando rotações no polígono inscrito na circunferência. Toussaint observa que diversos padrões rítmicos africanos são rotações uns dos outros e propõe os colares como um tipo de classificação dos ritmos. Como exemplo, apresentamos três importantes colares de ritmos africanos na figura 3.16.



**Figura 3.16:** Três classes de ritmos africanos de ciclos de doze unidades (Toussaint, 2003b)

Mostramos, nas subseções seguintes, os cálculos de outras propriedades dos padrões rítmicos com significados musicológicos mais claros do que as propriedades geométricas aqui apresentadas.

### 3.3.2 Igualdade de Espaçamento

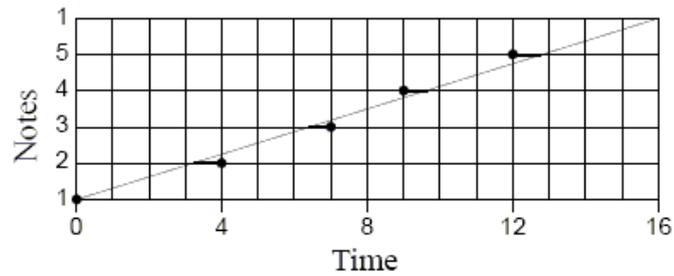
Segundo Toussaint (2003a) e London (2000), Clough e Duthett (1991) introduziram a noção de conjuntos maximamente espaçados<sup>5</sup> (*maximally even sets*) no estudo de notas das escalas representadas na circunferência. Posteriormente, Block e Douthett (1994) expandiram aquele trabalho criando uma medida de igualdade de espaçamento (*evenness*). A medida consistia em somar os tamanhos dos arcos formados por todos os possíveis pares de notas da escala. Toussaint (2003b) reutilizou esse conceito para o domínio dos padrões rítmicos para calcular igualdade de espaçamento entre ataques. Usando o modelo de representação poligonal dos ritmos, o cálculo é a soma dos tamanhos dos arcos formados entre pares de ataques. Toussaint (2003a) propôs também usar o tamanho das cordas (distância linear) ao invés dos tamanhos dos arcos, garantido melhor medição da igualdade de espaçamento.

Toussaint (2003a) apresenta, ainda, um outro método para calcular a igualdade de espaçamento, adaptado de uma medida de “*idealness*”<sup>6</sup> para escalas, proposto por Keith (1991), e que lembra o problema estatístico de regressão linear de um conjunto de pontos. Mostramos na figura 3.17 um exemplo do cálculo dessa medida para um padrão rítmico de cinco ataques, sobre uma métrica de dezesseis pulsos atômicos. Os ataques do padrão são os pontos destacados na figura 3.17, enquanto o segmento diagonal representa as posições “ideais” (para que os ataques fossem igualmente espaçados entre si). O cálculo da **medida de igualdade de espaçamento por regressão linear** consiste em somar as distâncias horizontais dos pontos à essa diagonal.

---

<sup>5</sup> Este é o mesmo conceito apresentado na seção 3.1 como requisito das métricas complexas.

<sup>6</sup> Toussaint (2003a) deixa a entender que Keith chama de *idealness* o que Clough e Duthett chamaram de *evenness*.



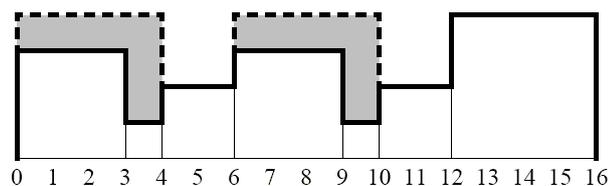
**Figura 3.17:** Igualdade de espaçamento por regressão linear

Analisando ritmos africanos de sete ataques baseados em ciclos de doze unidades, Toussaint (2003b) relata que as três famílias de ritmos que são mais difundidas (apresentadas na figura 3.17) são justamente as três configurações com maior igualdade de espaçamento. Além disso, dentre essas três famílias, a Bembé, que é a mais difundida, é justamente a família com maior igualdade de espaçamento.

### 3.3.3 Similaridade

Segundo Toussaint (2004a), uma medida de similaridade entre dois padrões rítmicos está presente no coração de qualquer algoritmo de comparação, reconhecimento ou classificação de ritmos. Toussaint (2004a) apresenta e compara várias medidas de similaridade de padrões rítmicos. Uma das medidas propostas é a **distância de Hamming**, que é o número de posições correspondentes diferentes nas representações TUBS dos padrões rítmicos. Outra medida é a **distância de permutação**, calculada como a soma das diferenças absolutas entre posições correspondentes dos vetores de ataques dos padrões. Já a **distância euclideana do vetor de intervalos** simplesmente aplica a medida de distância geométrica (distância Euclideana) aos vetores de intervalos dos padrões.

Outra medida de similaridade apresentada por Toussaint (2004a) é a **distância cronotônica**, computada como a área mínima delimitada pelas curvas das representações TEDAS de dois padrões rítmicos, como mostramos na figura 3.18.



**Figura 3.18:** Área mínima entre dois padrões rítmicos (doze, neste caso)

A distância cronotônica pode ser facilmente calculada subtraindo as posições correspondentes entre dois vetores cronotônicos e somando os valores absolutos dos resultados. Como o vetor cronotônico guarda os valores do gráfico TEDAS, isso equivale exatamente ao cálculo da área entre dois gráficos TEDAS. Na figura 3.19, apresentamos o cálculo da distância cronotônica para os padrões representados na figura 3.18.

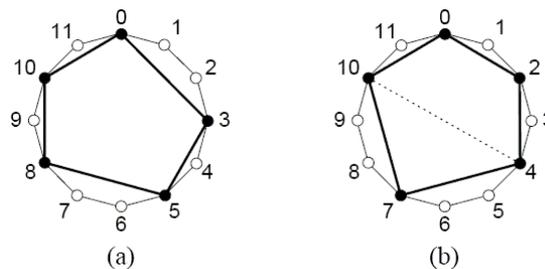
$$\begin{array}{r}
 \ominus \quad \begin{array}{cccccccccccccccc}
 4 & 4 & 4 & 4 & 2 & 2 & 4 & 4 & 4 & 4 & 2 & 2 & 4 & 4 & 4 & 4 \\
 3 & 3 & 3 & 1 & 2 & 2 & 3 & 3 & 3 & 1 & 2 & 2 & 4 & 4 & 4 & 4
 \end{array} \\
 \hline
 \Sigma \quad \begin{array}{cccccccccccccccc}
 1 & 1 & 1 & 3 & 0 & 0 & 1 & 1 & 1 & 3 & 0 & 0 & 0 & 0 & 0 & 0
 \end{array} = 12
 \end{array}$$

**Figura 3.19:** Cálculo da distância cronotônica usando os vetores cronotônicos

Analisando as medidas de similaridade aqui apresentadas, Toussaint (2004a) sugere que a distância cronotônica seja a mais fiel à percepção humana.

### 3.3.4 Imparidade Rítmica

Segundo Toussaint (2005), a **propriedade da imparidade rítmica** (*rhythmic oddity property*) foi proposta por Arom (1991) ao observar a cultura musical dos pigmeus africanos da etnia Aka. Uma padrão rítmico apresenta a propriedade em questão se ele não possuir nenhum par de ataques que divida o ciclo do padrão em bipartições de igual tamanho. Na figura 3.20 mostramos um padrão rítmico que apresenta a imparidade rítmica (padrão **a**) e outro padrão sem essa propriedade (padrão **b**). A reta tracejada indica a bi-partição que deixa o padrão **b** em desconformidade com a propriedade.



**Figura 3.20:** Padrão rítmico (a), usados pelos pigmeus africanos; padrão rítmico (b), usado no flamenco espanhol

Toussaint (2005) afirma que a imparidade rítmica pode ser considerada a “medida de preferência” usada pelos pigmeus para escolher seus ritmos<sup>7</sup>. Porém, analisando os padrões rítmicos africanos tocados em outras etnias, Toussaint observa que a mera análise da presença (ou ausência) dessa propriedade não é capaz de distinguir os padrões mais difundidos (entre as etnias africanas) dos menos difundidos. Por isso, ele propõe uma **medida de imparidade rítmica**, que é uma generalização da propriedade e consiste na contagem do número de possíveis bipartições iguais permitidas em um padrão. Quanto menos bipartições, maior a imparidade rítmica.

Toussaint afirma que a sua medida é capaz de medir as classes rotacionais de ritmos africanos que são mais difundidas. Na figura 3.16, mostramos como retas tracejadas as possíveis bipartições para cada classe rotacional de ritmos africanos. Nenhuma das classes apresenta a *propriedade* da imparidade rítmica, mas a *medida* de imparidade rítmica permite diferenciar a família mais difundida – a Bembé, com apenas uma bipartição possível.

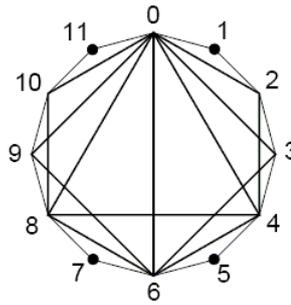
### 3.3.5 Off-Beatness

Também proposta por Toussaint (2003b), a medida de *off-beatness* consiste em contar a quantidade de pulsos “*off-beat*”. Na representação poligonal, pulsos *off-beat* são as posições do ciclo que não fazem parte de nenhum polígono regular inscrito na circunferência. Em uma definição mais algébrica, uma posição  $p$  de um ciclo de  $n$  unidades é considerada *off-beat* se  $p$  e  $n$  forem primos entre si.

Consideremos, como exemplo, ritmos de ciclos de doze unidades. O número doze possui divisão inteira apenas para quatro números inteiros maiores que um e menores que doze: seis, quatro, três e dois. Na representação poligonal dos ritmos esses números geram, respectivamente: um hexágono, um quadrado, um triângulo e um bi-ângulo inscritos na circunferência. A figura 3.21 apresenta as quatro posições do ciclo que não fazem parte de nenhuma dessas figuras. A medida de *off-beatness* de Toussaint consiste simplesmente em contar o número de ataques que ocorrem em alguma dessas posições.

---

<sup>7</sup> No Brasil, a propriedade da imparidade rítmica também é considerada o elemento mais comum às múltiplas manifestações do Samba (Sandroni, 1996).



**Figura 3.21:** Posições *off-beat*, segundo Toussaint

Toussaint (2003b; 2005) sugere que essa medida de *off-beatness* é melhor do que a medida de imparidade rítmica para discernir os padrões rítmicos africanos de maior difusão tanto dentro da cultura africana quanto internacionalmente.

### 3.3.6 Nível de Síncope

Alguns autores afirmam que emoções são induzidas pela música por meio de um contínuo processo de criação e relaxamento de tensões (Gómez et al., 2005). Essas tensões podem ser criadas em todos os níveis do fenômeno musical. Nessa perspectiva cognitiva, a síncope se destaca como um dos principais recursos de tensão no nível rítmico (temporal) da música. Em uma perspectiva mais musicológica, a síncope é vista como uma contradição momentânea da métrica.

Gómez et al. (2005) apresenta diversas medidas de síncope; porém duas delas não foram propostas originalmente com base na definição de síncope e já foram apresentadas nessa seção: a propriedade da imparidade rítmica (3.3.4) e a medida de *off-beatness* (3.3.5). Já as outras duas métricas apresentadas são tentativas reais de medir a sensação humana de síncope: a **medida de síncope de Keith** (1991); e a **distância ponderada da nota à batida** (*Weighted Note-to-Beat Distance* ou WNBD), proposta por aqueles autores.

Os autores compararam essas quatro medidas e chegaram à conclusão que a WNBD é a medida de síncope mais fiel à percepção humana. Além disso, a WNBD supera diversas desvantagens apresentadas pelas outras medidas tais como: inaplicabilidade em certas métricas e/ou em ritmos irregulares. O único problema, segundo o artigo, é a escolha dos pesos usados no cálculo da WNBD que ainda precisam ser definidos experimentalmente.

### 3.3.7 Complexidade

Segundo Shmulevich & Povel (2000), a noção de complexidade tem sido estudada no contexto da Teoria da Informação, sendo associada a conceitos como randomicidade, quantidade de informação e regularidade. No contexto musical, no entanto, o conceito de complexidade assume um caráter mais subjetivo, requerendo técnicas especializadas para a sua medição. Pressing (1997a) sugere associar complexidade rítmica com dificuldade de aprendizado, que, por sua vez, pode ser expressa por dificuldade de reconhecimento ou de reprodução.

Os artigos de Shmulevich & Povel (2000) e de Toussaint (2002) analisam diversas medidas de complexidade rítmica, algumas delas baseadas em técnicas da Teoria da Informação e outras baseadas em modelos da percepção musical humana. Analisando os resultados dessas medidas, eles sugerem duas como as mais fiéis à percepção humana: a **medida de Povel-Shmulevich** (PS-measure) e a **medida de Pressing** (1997a). Provavelmente não é coincidência o fato de ambas as medidas escolhidas serem baseadas em modelos da percepção rítmica humana, enquanto medidas da Teoria da Informação tiveram resultados menos relevantes.

Uma questão importante sobre o conceito de complexidade rítmica é que ele pode apresentar correlação com outras propriedades rítmicas aqui apresentadas, tais como como: nível de síncope, imparidade rítmica e (des)igualdade de espaçamento. Porém, essa correlação ainda não foi devidamente estudada na literatura.

## 3.4 Considerações Finais

Neste capítulo apresentamos, na primeira seção, um levantamento bibliográfico dos elementos estruturais rítmicos mais relevantes do ponto de vista cognitivo. Em seguida, analisamos diferentes alternativas de representação desses elementos estruturais rítmicos. Por fim, apresentamos algumas propriedades rítmicas calculadas a partir das diversas representações analisadas. Tais propriedades rítmicas podem servir para guiar a escolha dos padrões rítmicos em determinados contextos. No capítulo 5, mostramos como os diversos elementos aqui analisados foram usados na construção de um sistema rítmico multiagente. Antes, porém, mostraremos, no capítulo 4, o *framework* que serviu de base para construção do sistema.

## CAPÍTULO 4

# Framework para Aplicações Musicais

No capítulo 2, mostramos que sistemas multiagentes são inerentemente complicados, pois requerem uma certa infra-estrutura de software para permitir que os agentes operem e interajam. Os sistemas multiagentes do domínio musical requerem ainda outras características infra-estruturais como gerenciamento da saída de som e desempenho adequado. Neste capítulo apresentamos o *framework* desenvolvido para oferecer essa infra-estrutura para sistemas musicais, servindo de base para o sistema CInBalada que apresentaremos mais adiante neste trabalho.

Os dois objetivos principais assumidos na construção do *framework* foram: **padronização** e **desempenho**. A preocupação com desempenho se deve ao fato de que mesmo pequenos atrasos na execução sonora são facilmente percebidos por humanos, o que prejudicaria a execução sonora (em tempo real) do sistema. Já a preocupação com padronização se justifica pelo mesmo motivo que a literatura justifica a distribuição em rede<sup>8</sup> (cap. 2): para permitir a interação entre agentes de diferentes projetistas, criando situações mais ricas. A padronização completa dos sistemas musicais, no entanto, é um problema ainda em aberto na literatura. Por isso, apenas um esforço básico de padronização foi feito para a parte não-musical no *framework*.

Apesar de não ter sido exatamente um objetivo, o *framework* foi desenvolvido com desacoplamento do CInBalada, podendo ser usado para construir outras aplicações musicais multiagentes, mesmo aplicações não rítmicas. A primeira seção deste capítulo apresenta a plataforma multiagente do sistema (que é um *framework* por si só), enquanto a segunda seção aborda o agente gerenciador das saídas musicais.

---

<sup>8</sup> Não assumimos o objetivo de *distribuição em rede* por ele não ser necessário ao CInBalada e também porque acreditamos que a preocupação com padronização deve vir antes.

## 4.1 Framework Multiagente

O Framework para Aplicações MultiAgentes (**FAMA**) foi desenvolvido em conjunto com Rodrigo Lima, que já o apresentou em sua dissertação de mestrado (Lima, 2005). O FAMA é um *framework* de propósito geral, independente de domínio. Ele difere de outros *frameworks* multiagentes independentes de domínio (e.g. (Bellifemine et al., 1999) e (Poslad et al., 2000)) por considerar requisitos de desempenho, necessários em aplicações musicais e multimídias de um modo geral.

O FAMA pode ser visto de duas perspectivas: como uma **plataforma de execução** de agentes ou como uma **biblioteca de desenvolvimento** de sistemas multiagentes. A primeira perspectiva é apresentada na subseção 4.1.1 e considera a infraestrutura de tempo de execução que o FAMA oferece para agentes operarem. A segunda perspectiva, apresentada na subseção 4.1.2, considera as ferramentas oferecidas pelo FAMA para o desenvolvimento de sistemas multiagentes.

### 4.1.1 Plataforma de Execução

A plataforma do FAMA foi desenvolvida com base nas especificações da **FIPA** (*Foudation for Intelligent Physical Agents*; <http://www.fipa.org/>), que é uma organização internacional preocupada em difundir o paradigma multiagente, bem como em favorecer a interoperabilidade entre aplicações baseadas em agentes por meio da definição de padrões<sup>9</sup>. Dentre as diversas especificações disponibilizadas pela FIPA, as mais importantes para esse trabalho foram quatro (todas disponíveis no *site*):

1. *FIPA Abstract Architecture Specification* (SC00001)
2. *FIPA Agent Management Specification* (SC00023)
3. *FIPA Communicative Act Library Specification* (SC00037)
4. *FIPA ACL Message Structure Specification* (SC00061)

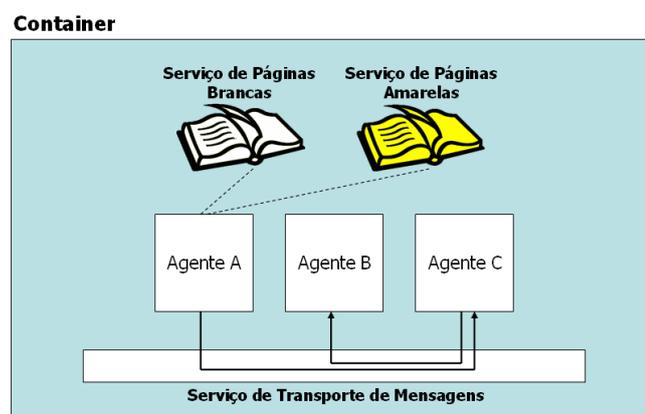
Essas especificações permitiram identificar os elementos básicos que viriam a compor a estrutura do *framework*. As duas primeiras especificações FIPA definem os serviços básicos que uma plataforma de execução de agentes deve oferecer. Baseada nessas duas especificações, a plataforma de execução do FAMA oferece os seguintes serviços básicos:

---

<sup>9</sup> Em 2005 a FIPA foi aceita como um comitê elaborador de padrões da IEEE Computer Society, mostrando a reputação que essa organização já alcançou na área.

- **Gerenciamento do ciclo de vida do agente.** A plataforma gerencia os possíveis estados do ciclo de vida dos agentes, bem como transições entre eles. No momento da criação dos agentes, eles são associados a identificadores únicos que servem de endereço para as mensagens trocadas.
- **Serviço de Transporte de Mensagens.** Repassa as mensagens enviadas por um agente aos seus destinatários. Provavelmente é o serviço mais importante numa plataforma multiagente, pois permite a interação entre os agentes.
- **Páginas Brancas.** Contêm descrições dos serviços providos por cada agente. Os serviços são identificados por um **tipo** (classificação) e um **nome** (específico). Um serviço de aluguel de filmes, por exemplo, poderia ser descrito com o tipo “locação” e o nome “filmes”.
- **Páginas Amarelas.** Fornecem aos agentes interessados os endereços dos agentes capazes de prover um serviço específico. O serviço pode ser identificado pelo tipo ou nome (pode ser ambos).

Semelhante ao que ocorre no framework de propósito geral JADE (Bellifemine et al., 1999), esses serviços são oferecidos no FAMA por um *container* de agentes (figura 4.1). Agentes desenvolvidos com a biblioteca de desenvolvimento do FAMA são automaticamente registrados no contêiner no momento da criação. O *container* assume, então, a responsabilidade pela operação (execução) dos agentes. Numa possível extensão do FAMA para oferecer distribuição, um *container* pode operar em cada nó da rede.



**Figura 4.1:** *Container* da plataforma FAMA

Uma questão que a FIPA não detalha é a modelagem dos mecanismos internos dos agentes. No FAMA, escolhemos modelar o funcionamento dos agentes por meio de

*behaviours*, novamente inspirado no JADE (Bellifemine et al., 1999). Cada *behaviour* representa um tipo de tarefa que o agente precisa executar. Fica a critério do desenvolvedor dos agentes a divisão do funcionamento geral do agente em *behaviours* distintos. Por exemplo: no domínio de comércio eletrônico, um agente vendedor pode ter um *behaviour* distinto para cada lidar com cada um dos agentes compradores com quem ele negocia.

**Tabela 4.1:** Algumas performativas implementadas no FAMA

<i>Performativa</i>	<i>Semântica</i>
INFORM	Informa que uma dada proposição é verdadeira.
REQUEST	Requisita ao agente destinatário que execute uma determinada ação, especificada na mensagem.
PROPOSE	Submete uma proposta para que o agente realize uma certa ação, satisfeita certas condições.
ACCEPT-PROPOSAL	Aceita uma proposta previamente submetida.
REJECT-PROPOSAL	Rejeita uma proposta previamente submetida.
CANCEL	Indica ao agente destinatário que não deseja mais a realização de uma certa ação.
SUBSCRIBE	Requisita ao agente destinatário para notificar de mudanças no valor de um objeto referenciado na mensagem.

Para a comunicação entre agentes, a FIPA se inspirou na Teoria dos Atos de Fala (Searle, 1969). Essa teoria vê uma mensagem (e.g. uma frase falada) como uma ação que possui uma intenção específica (e.g. informar, perguntar). Essa intenção é chamada de **performativa** e é considerada um elemento distinto (porém complementar) da mensagem propriamente dita. O uso de performativas em comunicação entre os agente serve para organizar a comunicação e reduzir ambigüidades. A terceira das especificações FIPA que citamos define as performativas disponíveis para comunicação entre os agentes. Na tabela 4.1, mostramos algumas das performativas incluídas no FAMA.

Já a quarta especificação FIPA usada como referência na construção do FAMA define a estrutura das mensagens trocadas entre os agentes. Essa especificação define os campos que devem estar presentes nas mensagens trocadas pelos agentes da plataforma. Mostramos, na tabela 4.2, os campos implementados no FAMA.

**Tabela 4.2:** Campos de mensagem implementados no FAMA

<i>Campo</i>	<i>Descrição</i>
sender	Identificador do agente emissor da mensagem.
receivers	Identificadores dos agentes aos quais a mensagem é dirigida.
Reply-to	Identificadores dos agentes aos quais devem ser dirigidas possíveis mensagens de resposta (quando não especificado, a resposta é encaminhada ao <i>sender</i> da mensagem corrente).
performative	Intenção da mensagem.
content	A mensagem propriamente dita.
language	Este campo serve informar qual a mensagem usada no conteúdo da mensagem (uma vez que a FIPA não obriga o uso de uma linguagem específica) .
conversation-id	Identifica o fluxo de mensagens do qual a mensagem faz parte.

Agora que apresentamos a plataforma multiagente do FAMA, veremos na próxima subseção como ela foi implementada para servir como uma biblioteca de desenvolvimento de sistemas multiagentes.

#### 4.1.2 Implementação

O FAMA foi desenvolvido em C++, por questões de desempenho. Na implementação deste *framework*, utilizamos como referência o JADE (Bellifemine et al., 1999), desenvolvido em Java, que é talvez o mais importante *framework* independente de domínio para sistemas multiagentes.

Na figura a 4.2 mostramos o diagrama de classes do FAMA. As duas classes mais importantes oferecidas para o desenvolvimento de sistemas multiagentes são: *FmAgent* e *FmBehaviour*<sup>10</sup>. Um agente do FAMA é criado como uma especialização da classe *FmAgent* e deve possuir pelo menos um *behaviour*, implementados como especializações de *FmBehaviour*. Uma especialização de *FmBehaviour* deve definir uma ação não bloqueante (sem laço infinito) e uma condição de encerramento do *behaviour*. Alternativamente, um *behaviour* pode ser implementado como especialização de

---

<sup>10</sup> Usamos o prefixo “Fm” como padrão de nomenclatura para as classes do *framework* musical.

FmOneShotBehaviour (executa apenas uma ação e encerra) ou de FmCyclicBehaviour (executa suas ações indefinidas vezes).

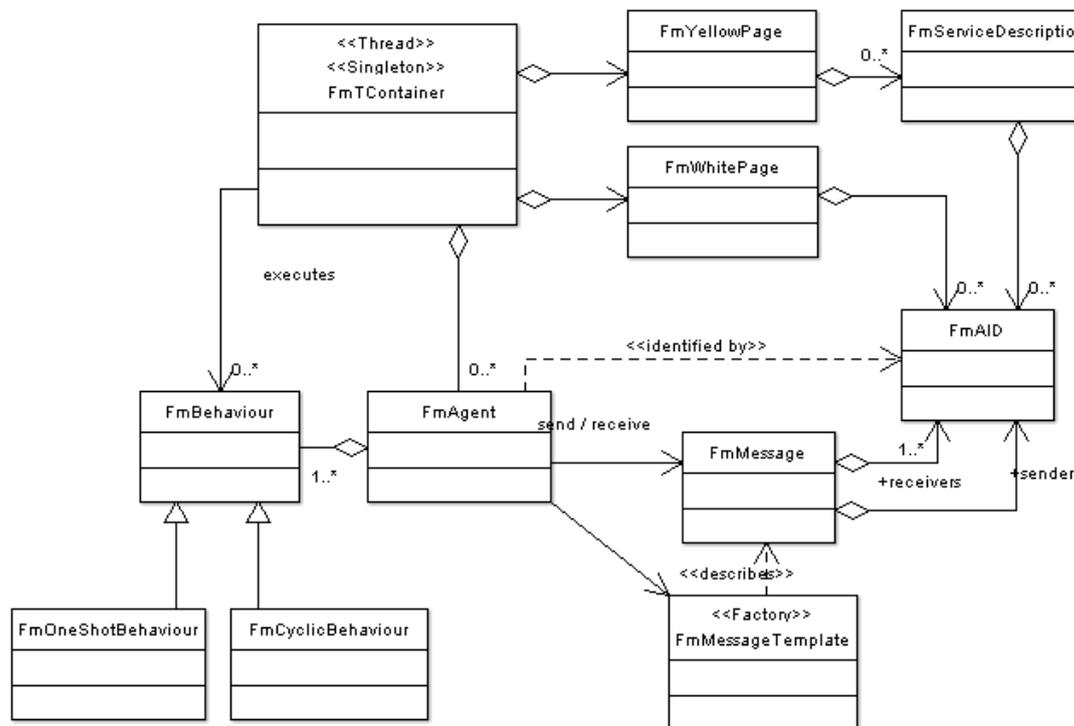
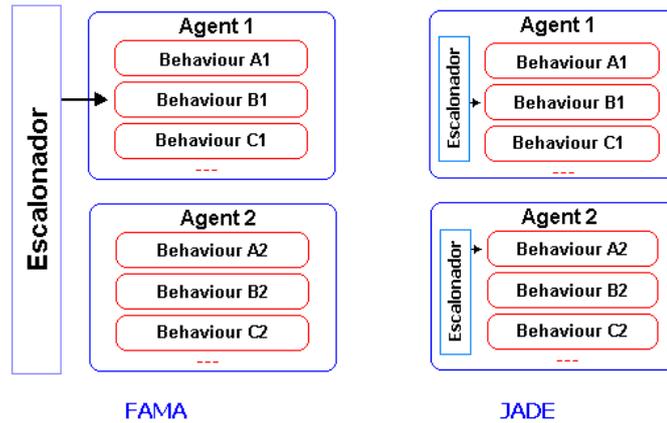


Figura 4.2: Diagrama de classes do FAMA

Os serviços da plataforma FAMA (i.e. gerenciamento do ciclo de vida dos agentes, serviço de mensagens e páginas brancas e amarelas) são disponibilizados aos agentes de maneira centralizada pela classe `FmTContainer`. Ao ser criado, o agente (instância de uma subclasse de `FmAgent`) automaticamente se registra na instância única de `FmTContainer`, permitindo que o agente registre seus *behaviours* e, assim, inicie suas atividades na plataforma. Quando o agente indicar seu encerramento, caberá ao *container* a desalocação do objeto.

Inspirado no JADE, o *container* do FAMA é também o responsável por executar os *behaviours* dos agentes, permitindo a atuação destes na plataforma. Porém, no FAMA, os *behaviours* são executados centralizadamente (*round robin*) na *thread* única do *container*, enquanto no JADE cada agente é uma *thread* separada que escalona seus *behaviours* independentemente (figura 4.3). Com apenas uma *thread* para toda a plataforma, acreditamos que o desempenho do FAMA seja favorecido.



**Figura 4.3:** Escalonamento de *behaviours* no FAMA e no JADE

Os agentes são identificados unicamente por meio de objetos do tipo `FmAID`. Esse identificador do agente é atribuído automaticamente pelo *container* na criação do agente e pode ser consultado a partir do nome do agente por meio do serviço de páginas brancas, implementando pela classe `FmWhitePage`. O identificador de um agente também pode ser obtido a partir do serviço prestado por ele no serviço de páginas amarelas, implementado pela classe `FmYellowPage`.

Os agentes se comunicam trocando objetos da classe `FmMessage`, que implementa o formato de mensagem apresentado na subseção anterior. O envio e recebimento de mensagens se dá por meio da classe `FmTContainer`. Emissor e destinatários da mensagem são identificados na mensagem pelos respectivos objetos do tipo `FmAID` de cada agente. Já a classe `FmTemplate` serve para descrever propriedades das mensagens e podem ser usadas pelos agentes para um recebimento seletivo de mensagens. Por exemplo, um objeto dessa classe pode ser criado para selecionar apenas as mensagens com performativa `INFORM` e com `conversation-id` “`main_flow_3`”.

Assim, FAMA serve como um *framework* para sistemas multiagentes independente de domínio. Porém, como dissemos no início deste capítulo, ele é parte de um *framework* maior para aplicações musicais. Na próxima seção veremos a segunda parte desse *framework* maior, especificamente voltada para o gerenciamento da saída sonora em sistemas musicais multiagentes.

## 4.2 Agente Mixer

A descentralização do processamento é inerente aos sistemas multiagentes. Essa descentralização traz alguns problemas para o uso de certos recursos disponíveis de

maneira centralizada. Nos sistemas musicais, o recurso tipicamente centralizado que precisa ser gerenciado é a saída sonora do sistema, seja ela em arquivo ou placa de som.

No *framework* para aplicações musicais multiagentes aqui apresentados, o **agente *mixer*** (desenvolvido sobre o FAMA) é o responsável por gerenciar centralizadamente toda a saída musical gerada pelos agentes do sistema. O uso de um agente centralizador das saídas musicais é relativamente comum em sistemas musicais multiagentes (e.g. (Murray-Rust, 2003) e (Costalonga et al., 2003)), pois apresenta diversas vantagens:

- **Modularização.** Com a separação entre criação e execução dos sons, um desenvolvedor de agentes musicais precisa se preocupar apenas com questões relacionadas à composição musical.
- **Reuso.** Ao mesmo tempo, a modularização permite que o agente *mixer* seja independente de aplicação e possa ser reusado, uma vez que ele independe do processo de criação utilizado.
- **Saída sonora unificada.** O agente *mixer* serve como o ponto único de execução sonora do sistema, facilitando a sincronização entre as saídas dos agentes e permitindo que a performance seja salva ou ouvida por inteiro.

O agente *mixer* usa o formato MIDI nas saídas sonoras, devido à ampla difusão desse padrão e devido ao equilíbrio entre facilidade de manipulação simbólica e fidelidade de reprodução em relação à performance original que ele oferece (conforme Murray-Rust (2003) e Dannenberg (1993), por exemplo). São duas as saídas sonoras MIDI que o agente *mixer* oferece:

- **Placa de Som.** É a saída audível em tempo real do sistema. O mixer utiliza um temporizador interno para enviar, no momento adequado, cada nota para a saída MIDI. Na verdade, na maioria dos sistemas computacionais atuais, existem várias possíveis saídas MIDI oferecidas pela placa de som ou pelo sistema operacional. Cada uma delas realiza a síntese das notas de maneira diferente. O agente *mixer* do CInBalada pode ser configurado para usar qualquer das saídas MIDI disponíveis.
- **Arquivo MIDI.** Registra a performance completa de todos os agentes, permitindo que ela seja editada e, possivelmente, reaproveitada pelo usuário do sistema em alguma composição MIDI maior.

Além do controle da saída de som, o agente *mixer* também oferece algumas funcionalidades básicas de controle da performance dos agentes:

- **Mudança da métrica.** Para o agente *mixer* interessa apenas o número de pulsos atômicos da métrica. Essa informação basta para manipular os trechos musicais recebidos. Os agentes musicais se ajustam a eventuais mudança no *mixer*, mandando apenas padrões rítmicos que sejam compatíveis.
- **Mudança de andamento** (em BPM). É refletida em tempo real no processo de seqüenciamento para a placa de som e também é registrada no arquivo MIDI. É imperceptível para os agentes musicais do sistema.
- **Início e interrupção da performance** - *play*, *pause* e *stop*. A performance é iniciada ou retomada com um comando *play*. A partir de então, ela pode ser interrompida com o comando *stop* ou apenas temporariamente interrompida com o comando *pause* (podendo ser retomada do ponto em que parou).
- **Resolução do seqüenciador.** Define a precisão do *timer* usado para seqüenciar as frases dos agentes. Quanto mais alta a resolução, mais fielmente as saídas reproduzirão os desvios temporais expressivos das frases (caso haja).

#### 4.2.1 Ciclo de Vida

Quando o agente *mixer* é criado na plataforma FAMA, ele se registra como provedor do serviço de tipo "*Music Output*" e nome "*MIDI Output*". Porém, se ele identificar que já existe algum agente registrado nas páginas amarelas com esse serviço, ele encerra sua execução. Apenas se ele for o único agente do serviço em questão é que ele começa a operar. Uma vez em operação, o agente *mixer* só é encerrado quando a plataforma FAMA é finalizada.

Durante sua operação, o *mixer* aguarda que novos agentes musicais enviem mensagens para se cadastrar no seu serviço. Uma vez cadastrado, o agente musical será notificado regularmente da necessidade de enviar ao *mixer* novas frases (de duração de um compasso) para serem encaminhadas às saídas. A notificação indica o número do compasso da performance global. Quando o *mixer* recebe uma frase, ele a grava-a tanto em uma pista (*track*) exclusiva do agente em um arquivo MIDI central, como em certas estruturas temporárias usadas para seqüenciar as frases para uma saída MIDI da placa de

som. Quando o compasso está próximo do fim, o agente *mixer* novamente notifica os agentes do próximo compasso reiniciando o processo.

Mostramos na tabela 4.3 as possíveis mensagens trocadas (recebidas ou enviadas) pelo *mixer* com outros agentes, indicando a performativa e o conteúdo da mensagem. As duas primeiras mensagens são específicas do agente *mixer* e, portanto, formam uma linguagem própria chamada **mixer-agent-sl**. A terceira mensagem é usada em comunicações entre quaisquer agentes do sistema para representar um trecho musical e, portanto, é representada em uma linguagem independente do *mixer* chamada **fama-music-xml**.

**Tabela 4.3:** Mensagens usadas pelo agente *mixer*

<i>Significado</i>	<i>Performativa</i>	<i>Conteúdo</i>	<i>Linguagem</i>
Agente musical deseja começar a usar a saída	SUBSCRIBE	midi output service	mixer-agent-sl
Mixer sinaliza a ocorrência de um novo compasso	INFORM	bar=[número]	mixer-agent-sl
Agente musical informa o trecho musical do compasso	INFORM	[trecho-musical]	fama-music-xml

O parâmetro da mensagem de notificação de um novo compasso (tabela 4.3) é simplesmente a posição inteira do compasso na performance global, a partir de zero. Já o trecho musical, conforme já dito, é especificado em **fama-music-xml**, que representa informações de performance (análogas à MIDI) em sintaxe XML. A escolha de XML foi motivada pela facilidade de manipulação e interpretação dessa linguagem. Na figura 4.4 apresentamos um exemplo de frase especificada nesse formato.

```
<Phrase resolution="400" patch="12">
  <Note pitch="36" dynamic="80" startTick="12" endTick="18"/>
  <Note pitch="43" dynamic="80" startTick="30" endTick="45"/>
</Phrase>
```

**Figura 4.4:** Exemplo de trecho musical em *fama-music-xml*

Na próxima subseção, mostramos como o agente *mixer* foi implementado usando o FAMA.

### 4.2.2 Implementação

Na figura 4.5 apresentamos o diagrama com as classes que implementam o agente *mixer*. Em branco estão representadas as classes do FAMA, enquanto as classes mais escuras são específicas do agente *mixer*.

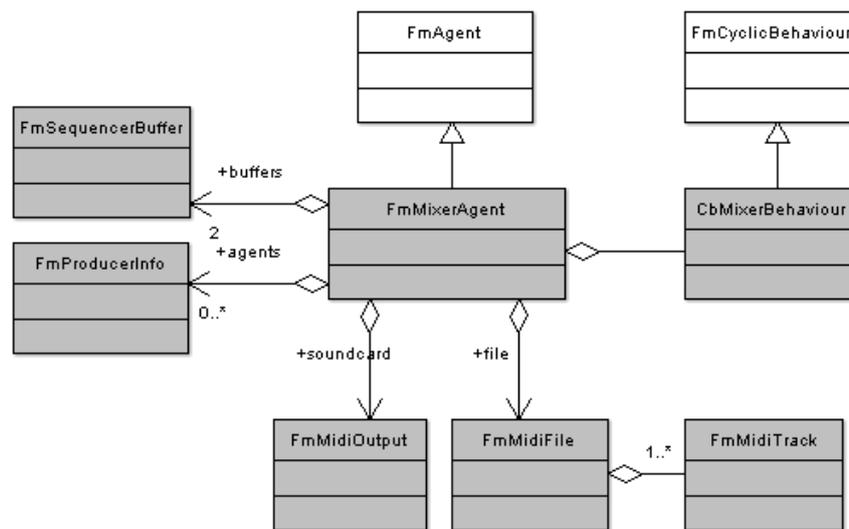


Figura 4.5: Classes que compõem o agente *mixer*

O agente *mixer* é representado basicamente pela classe `FmMixerAgent`, que é uma especialização da classe `FmAgent` do FAMA. Essa classe oferece os métodos que implementam as funcionalidades de controle da performance citadas no começo da seção. O agente possui um único *behaviour*, implementado pela classe `CbMixerBehaviour`. Esse *behaviour*, que é executado ciclicamente (especialização de `FmCyclicBehaviour`), implementa o protocolo de comunicação com agentes músicos que foi apresentado na subseção anterior.

Quando um agente se registra no agente *mixer* como compositor (produtor), suas informações ficam cadastradas na classe `FmProducerInfo`. Já as frases enviadas pelos agentes devidamente cadastrados são armazenadas na classe `FmSequencerBuffer`, a partir da qual o agente *mixer* seqüencia para as classes `FmMidiOutput` (placa de som) e a `FmMidiFile` (arquivo MIDI). A classe `FmMidiOutput` permite que seja escolhida qualquer uma das saídas MIDI oferecidas pela placa de som (para síntese por *hardware*) ou pelo sistema operacional (síntese por *software*).

### **4.3 Considerações Finais**

Neste capítulo, apresentamos um *framework* para desenvolvimento de aplicações musicais baseadas em agentes inteligentes. O *framework* proposto é dividido em dois módulos: um módulo para criação de sistemas multiagentes (o FAMA) e um módulo para gerenciamento das saídas sonoras do sistema (o agente *mixer*). No próximo capítulo, mostramos a modelagem do sistema CInBalada feita sobre a infra-estrutura apresentada.

## CAPÍTULO 5

# Modelagem do CInBalada

Conforme visto no capítulo 2, existem poucos sistemas de composição rítmica polifônica na literatura, todos baseados em uma das duas técnicas: algoritmos genéticos e sistemas multiagentes. Comparando as duas técnicas, entendemos que a abordagem multiagente apresenta a vantagem de permitir uma modelagem mais natural de uma performance em grupo, além de permitir a geração de saídas musicais em tempo real (enquanto algoritmos genéticos geralmente requerem longas execuções). Porém, dos sistemas multiagentes encontrados, apenas o Rhythm Band Editor (Pachet, 2000) permite realmente o uso de múltiplos agentes rítmicos.

Neste capítulo apresentamos o sistema CInBalada para composição rítmica polifônica. O sistema usa uma abordagem alternativa de aplicação do paradigma multiagente para o problema da composição rítmica polifônica. Enquanto no Rhythm Band Editor os agentes partiam de um único padrão rítmico inicial que podia ser alterado segundo regras individuais dos agentes, no CInBalada cada agente possui um conjunto de *padrões rítmicos prontos* que o agente pode escolher para tocar (sem alterações). Além disso, diferentemente do Rhythm Band Editor que produzia performances em grupo sem interação direta entre os agentes, no CInBalada as performances são resultado de um processo de *negociação entre os agentes* participantes.

O enfoque da presente pesquisa também difere do Rhythm Band Editor. Enquanto o objetivo do Rhythm Band Editor era estudar as regras de transformação e seus efeitos nas performances polifônicas resultantes, o objetivo do CInBalada é servir como um ambiente para experimentação rítmica a ser usado por músicos para auxiliar na composição de seções rítmicas novas. Para garantir que sistema fosse realmente útil para músicos humanos, consideramos como objetivo a ser perseguido a geração de seções rítmicas que atendam aos seguintes requisitos (Pease et al., 2001):

- a. **Qualidade musical.** As saídas do sistema devem ser aproveitáveis como partes de composições humanas maiores (e.g. acrescentadas de harmonia

e melodia), ou seja, devem ter um valor estético o mais próximo possível ao de composições humanas.

- b. **Diversidade.** O sistema deve apresentar uma certa variedade de resultados entre execuções distintas do sistema. Essa diversidade deve ser oferecida por meio de parâmetros de personalização das composições, permitindo que o sistema gere uma diversidade controlada.

Assumindo os requisitos acima, apresentamos no restante desse capítulo a modelagem conceitual do sistema CInBalada, construído usando as técnicas de manipulação musical discutidas no capítulo 3. Na seção 5.1, falamos dos problemas encontrados nas abordagens preliminares e como eles nos levaram a propor a noção de *papéis rítmicos*, detalhada na seção 5.2. Na seção 5.3, apresentamos uma visão geral da arquitetura do sistema, proposta com base nos *papéis rítmicos*. Os detalhes da arquitetura são apresentados em cada uma das seções posteriores: armazenamento dos padrões, na seção 5.4; as representações da estrutura rítmica, na seção 5.5.; o mecanismo de avaliação dos padrões, na seção 5.6; e o funcionamento detalhado dos agentes percussionistas na seção 5.7.

## **5.1 Abordagens Preliminares**

Antes de apresentar a modelagem final do sistema CInBalada, mostraremos nesta seção as primeiras tentativas de desenvolvimento do sistema e como elas conduziram à abordagem final apresentada nas seções seguintes. Desde a primeira versão, o sistema CInBalada foi baseado no paradigma multiagente (Sampaio, 2003). Cada agente do sistema era conhecedor de um repertório pré-definido de padrões rítmicos de um instrumento de percussão específico. Desse modo, o sistema evitava o problema maior de criar performances polifônicas nota por nota, para basear o sistema na rearrumação de frases prontas.

Jennings et. al. (2001) afirmam que interações entre agentes são parte central de sistemas multiagentes e que o mecanismo mais poderoso de interação entre os agentes é a *negociação* – o processo pelo qual um grupo de agentes tenta atingir um acordo mutuamente aceitável. Por esse motivo, o CInBalada foi também concebido como um sistema baseado em negociação multiagente desde a sua primeira versão. Portanto, o objetivo principal da primeira versão do sistema era validar o uso de negociação multiagente para a geração de composições rítmicas polifônicas a partir de padrões rítmicos monofônicos.

O processo de negociação ocorria apenas quando um novo agente era inserido no sistema. A partir de então, o agente novato mostrava cada um dos seus padrões rítmicos ao restante dos agentes, até que um dos padrões fosse aceito pelo grupo. Isso acontecia apenas quando todas as possíveis combinações de pares de padrões (considerando um padrão de cada agente) fossem consideradas aceitáveis segundo uma função de avaliação única e um limiar de aceitação. Caso nenhum padrão do agente novato satisfizesse essas condições, o agente não seria aceito no grupo e encerraria sua execução.

Essa abordagem apresentava alguns problemas, diante dos objetivos propostos. O primeiro problema era que, após a escolha dos padrões, os agentes permaneciam em ciclo tocando o mesmo padrão a cada compasso, pois o sistema não previa qualquer mecanismo de variação temporal do padrão. Isso poderia ser pouco interessante aos usuários alvo do sistema. Outra situação pouco desejável aos usuários era a possibilidade de um agente inserido no sistema ser removido sem chegar a tocar com o resto do grupo, por conta do mecanismo de negociação adotado.

O uso de uma função de avaliação única para todos os agentes também trazia alguns problemas. O primeiro seria um problema conceitual, pois fazia com que o uso da abordagem multiagente fosse questionável (bastava usar a função junto com algum algoritmo de busca, por exemplo). Além disso, era questionável a possibilidade de concentrar em uma única medida de avaliação a noção de qualidade musical. Como qualidade musical é um conceito subjetivo, o ideal seria dar ao usuário algum controle musical sobre essa avaliação, o que não era possível usando uma única medida de avaliação.

O primeiro passo para tentar resolver os problemas encontrados foi pesquisar diversas medidas alternativas de avaliação, como pode ser conferido na seção 3.3. Porém, ainda permanecia a dificuldade de organizar esse conhecimento musical, de modo a oferecer parâmetros de alto-nível para personalização das saídas musicais do sistema. O conceito que propomos na versão final do CInBalada para organizar as medidas de avaliação do sistema é apresentado na seção seguinte.

## **5.2 Papéis Rítmicos**

Analisando as batucadas de samba, que são performances puramente rítmicas características do carnaval brasileiro, Salazar (1991) divide informalmente os instrumentos de percussão em grupos relacionados. Para cada grupo, Salazar descreve de

maneira geral a função assumida por qualquer daqueles instrumentos durante uma batucada de samba.



**Figura 5.1:** Partituras de alguns instrumentos de uma batucada de samba (Salazar, 1991)

As principais funções dos instrumentos de batucada de samba, com algumas pequenas modificações em relação à classificação informal de Salazar<sup>11</sup>, feitas com base em observações das partituras das linhas instrumentais:

- **Surdo:** instrumento base, responsável por manter o ritmo. Observando alguns exemplos de batucada, vemos que o surdo procura demarcar os pulsos da métrica (vide figura 5.1) servindo, portanto, como referência principal para todos os demais instrumentos dentro da batucada.
- **Caixa, Ganzá, Pandeiro, Reco-Reco e Congas:** têm mais liberdade para fazer variações e complementam, junto com o surdo, a base rítmica do samba. Os ataques de suas linhas são relativamente bem espaçados e ressaltam pulsos de menor granularidade (pulsos atômicos), preenchendo a performance com maior quantidade de ataques (vide figura 5.1).
- **Tamborim, Agogô, Cuíca e Apito:** instrumentos solistas que têm grande liberdade para fazer variações. Os ataques de nota acontecem de maneira

---

<sup>11</sup> As congas aqui são consideradas como instrumentos de complementação da base rítmica (Salazar as tratava como um grupo à parte) e o apito como um instrumento solista (que Salazar também tratava à parte por ser o instrumento usado pelos maestros).

mais irregular (mal espaçados), deixando a performance mais dinâmica e interessante (vide figura 5.1).

Portanto, cada um dos grupos apresentados se caracteriza por assumir papéis distintos na performance. Como estamos lidando com performances puramente rítmicas, chamaremos esses papéis de **papéis rítmicos**. Aos três papéis rítmicos da batucada de samba identificados acima chamaremos de: **base** (o surdo), **base complementar** (caixa e demais instrumentos do seu grupo) e **solo** (tamborim, agogô e cuíca).

Apesar de terem sido aqui observados apenas nas batucadas de samba, esses mesmos papéis também aparecem em outras manifestações musicais. Por exemplo, o papel *base* também aparece em diversos gêneros latinos, sendo executados por um par de claves (blocos de madeira) ou por um cowbell (Toussaint, 2002). Nessas manifestações, o papel *base* toca ritmos clave, que têm a mesma função de uma métrica conforme mostramos na seção 3.1. Também o papel rítmico *solo* é bastante difundido, sendo amplamente encontrado na música popular de um modo geral<sup>12</sup>.

Nas batucadas de samba, como em muitos outros gêneros musicais, há situações em que os instrumentos tocam certos padrões rítmicos curtos que sinalizam uma quebra ou mudança no curso da performance do grupo. Essas situações são chamadas de **viradas** e podemos considerá-las como um tipo de papel rítmico temporário (após executar uma virada, os instrumentos geralmente retornam ao papel anterior). Isso é um claro exemplo de que a função do instrumento pode variar durante a execução. Por isso, assumimos que qualquer papel rítmico pode variar durante a performance. Por exemplo, um certo instrumento pode iniciar com o papel *base complementar*, depois assumir o papel *virada* por poucos compassos e, em seguida, passar para o papel *solo*.

Além dos papéis rítmicos das batucadas de samba identificados acima, outros papéis podem ser identificados analisando outras manifestações musicais. Relacionamos aqui apenas mais dois papéis encontrados durante a pesquisa, e que são derivados da relação de **chamada** e **resposta** (vide seção 3.1). Quando as frases de chamada e de resposta são executadas em instrumentos distintos, assumimos que são dois papéis diferentes, porém relacionados entre si. O padrão rítmico executado pelo instrumento de resposta é uma repetição ou um complemento do padrão do papel de chamada e, em geral, há pouca ou nenhuma sobreposição temporal entre eles.

---

<sup>12</sup> Apesar de, nesse contexto, estarmos nos referindo a instrumentos de percussão, mesmo os solos realizados em instrumentos melódicos também podem ser considerados como sendo do papel rítmico solo.

### 5.2.1 Elementos dos Papéis Rítmicos

De maneira simplificada, os papéis rítmicos definem quais são os padrões rítmicos mais adequados em um dado contexto de performance. Essa adequação é avaliada por meio dos seguintes elementos básicos que compõem cada possível papel rítmico:

- a. **Critério de elegibilidade dos padrões rítmicos individuais.** Define as características individuais que os padrões rítmicos do papel devem apresentar. Por exemplo: o papel *base* deve tocar padrões de poucos ataques bem-espaçados.
- b. **Papel de referência.** O papel de referência influencia os padrões que podem ser escolhidos pelo instrumento. Isso cria uma hierarquia entre os papéis, em que certos papéis são subordinados a outros. Alguns papéis, no entanto, não possuem papel de referência. Por exemplo, o padrão rítmico tocado por um instrumento do papel *base complementar* vai depender do padrão tocado pelo instrumento *base* (que, por sua vez, não depende de nenhum outro papel).
- c. **Critério de adequação ao papel de referência.** Esse critério define como o papel de referência vai influenciar a escolha do padrão rítmico (critério vertical). Um exemplo de critério seria: todos os ataques do papel *base* devem ser ataques acentuados do papel *base complementar*.
- d. **Critério de coesão do papel rítmico.** Avalia, de maneira global, a qualidade da combinação entre os padrões rítmicos tocados pelos instrumentos de um mesmo papel (critério horizontal). Por exemplo, dois instrumentos do papel *virada* precisam executar padrões similares para que, conjuntamente, permaneçam fiéis ao caráter do papel.
- e. **Política de evolução dos padrões.** Define a liberdade de variação dos padrões, com base no histórico dos últimos padrões escolhidos. Ela define se o padrão será repetido ciclicamente sem alterações ou se o papel poderá executar variações do padrão. O papel *solo*, por exemplo, pode variar o padrão rítmico a cada compasso.

Dentre os elementos, quatro critérios de avaliação estão presentes (**a**, **c**, **d**, **e**), enquanto o outro elemento (**b**) estabelece uma relação de dependência entre os instrumentos. Dos quatro critérios, apenas o critério **a** leva em consideração propriedades individuais do padrão, enquanto os demais critérios usam informações do contexto da performance, tais como: padrões tocados por instrumentos de outros papéis e o histórico

da performance. Por esse motivo, chamaremos estes três últimos elementos de *critérios contextuais*.

Considerando essa definição de papéis rítmicos, explicamos, na próxima subseção, uma proposta de como podem ser caracterizadas as diferentes manifestações musicais rítmicas encontradas em diversas culturas (e.g. batucadas de samba, maracatu).

### 5.2.2 Manifestações Rítmicas

Com base no conceito de papéis rítmicos, podemos dizer que manifestações musicais rítmicas polifônicas (como as batucadas de samba) podem ser caracterizadas pelos seguintes elementos:

- **Orquestração.** Tipos de instrumentos e quantidades de cada instrumento. Também podemos considerar a técnica de execução do instrumento (e.g. acentuações) como dependente da manifestação musical.
- **Papéis rítmicos.** Os papéis rítmicos que estão presentes na manifestação e quais instrumentos assumem cada um desses papéis.
- **Repertório.** Os padrões rítmicos recorrentes na manifestação rítmica e que caracterizam estilisticamente a manifestação.

Observamos que o último elemento não é diretamente dependente dos papéis rítmicos, pois os papéis definem apenas características gerais que os padrões devem apresentar. Por exemplo, o papel rítmico *base* no samba (assumido pelo surdo) executa padrões rítmicos diferentes do mesmo papel *base* na salsa (assumido pelas claves ou pelo cowbell), mas tanto o padrão rítmico de samba como o padrão de salsa satisfazem o critério de elegibilidade do papel *base*.

Na próxima subseção, apresentamos uma análise de como os conceitos propostos podem servir para a construção de um sistema rítmico.

### 5.2.3 Papéis Rítmicos em Sistemas de Composição

Como visto anteriormente, um papel rítmico é um conjunto de critérios que controla a escolha dos padrões rítmicos de um instrumento num dado contexto. Numa perspectiva computacional, o conceito de papel rítmico pode servir como um mecanismo descentralizado (por ser aplicado instrumento a instrumento) de avaliação de performances rítmicas polifônicas.

Individualmente, cada um dos critérios que compõe os papéis rítmicos, tem uma função bem definida e pode ser implementado usando as medidas de propriedades

rítmicas apresentadas no capítulo 3. O conceito aqui proposto serve, portanto, para **organizar o conhecimento musical** analisado naquele capítulo.

Como vimos na seção 3.1, a música possui uma estrutura subjacente que é percebida pela mente humana e é responsável pelo valor estético atribuído a ela. Portanto, como resultado da organização imposta pelo conceito papéis rítmicos a uma performance rítmica, acreditamos que esse conceito possa permitir que se alcancem resultados de boa **qualidade musical** em um sistema de composição automática.

Além disso, um sistema de composição rítmica baseado em papéis rítmicos pode facilmente permitir a parametrização dos três elementos que caracterizam as manifestações rítmicas – orquestração, distribuição dos papéis rítmicos entre os instrumentos e o repertório de padrões rítmicos. Como resultado, o sistema poderia gerar uma grande **diversidade** de composições, inclusive composições **novas**, que rompam as barreiras estilísticas de uma manifestação rítmica específica.

Pelos motivos aqui apresentados, o conceito de papéis rítmicos foi usado na modelagem final do sistema CInBalada. A partir da próxima seção, mostramos os detalhes da modelagem.

### **5.3 Arquitetura do Sistema**

O sistema CInBalada baseia-se no paradigma multiagente para gerar performances rítmicas polifônicas. Considerando as diferentes abordagens apresentadas na seção 2.2, o sistema usa aquela em que cada agente representa um músico de uma performance em grupo. No caso do CInBalada, os agentes musicais tocam apenas instrumentos de percussão e podem assumir papéis rítmicos diversos no decorrer da performance.

Como já comentamos antes, um dos principais motivos para o uso do paradigma multiagente é que ele favorece a construção de sistemas musicais, já que um grupo musical é um sistema inerentemente distribuído (cada músico podendo ser considerado um agente). Outro motivo é que ele é adequado para implementar papéis rítmicos, já que eles funcionam como um mecanismo *descentralizado* de avaliação conforme comentamos no final seção anterior.

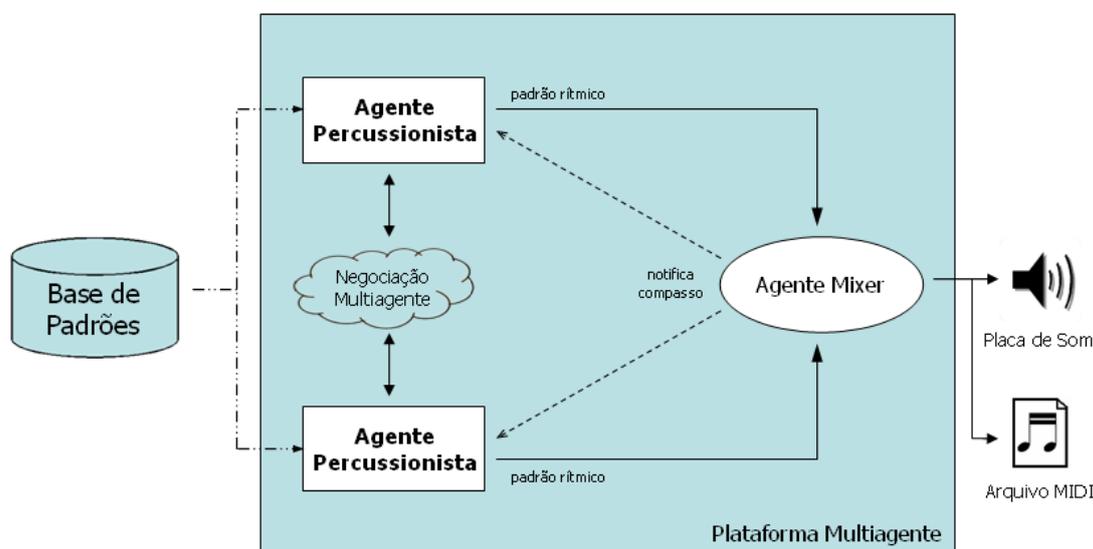
O sistema CInBalada permite a criação de **projetos rítmicos**, que é o nome dado às possíveis configurações dos seguintes parâmetros do sistema:

- **Orquestração.** Os instrumentos são escolhidos livremente pelo usuário (dentre as opções oferecidas pelo sistema).

- **Papéis rítmicos.** Inicialmente o usuário escolhe um único papel rítmico *default* para cada instrumento, mas o sistema também permite que o usuário altere independentemente o papel rítmico do instrumento para cada compasso.
- **Outros parâmetros básicos.** A métrica, o andamento e a duração (em compassos) da performance, todos podem ser escolhidos livremente pelo usuário.

Os projetos rítmicos são criados pelos usuários e servem como entrada para o sistema gerar performances rítmicas polifônicas. Execuções distintas de um mesmo projeto rítmico podem gerar composições diferentes, porém com características similares (fiéis à especificação do projeto). Já as composições geradas por projetos rítmicos diferentes são mais distintas entre si.

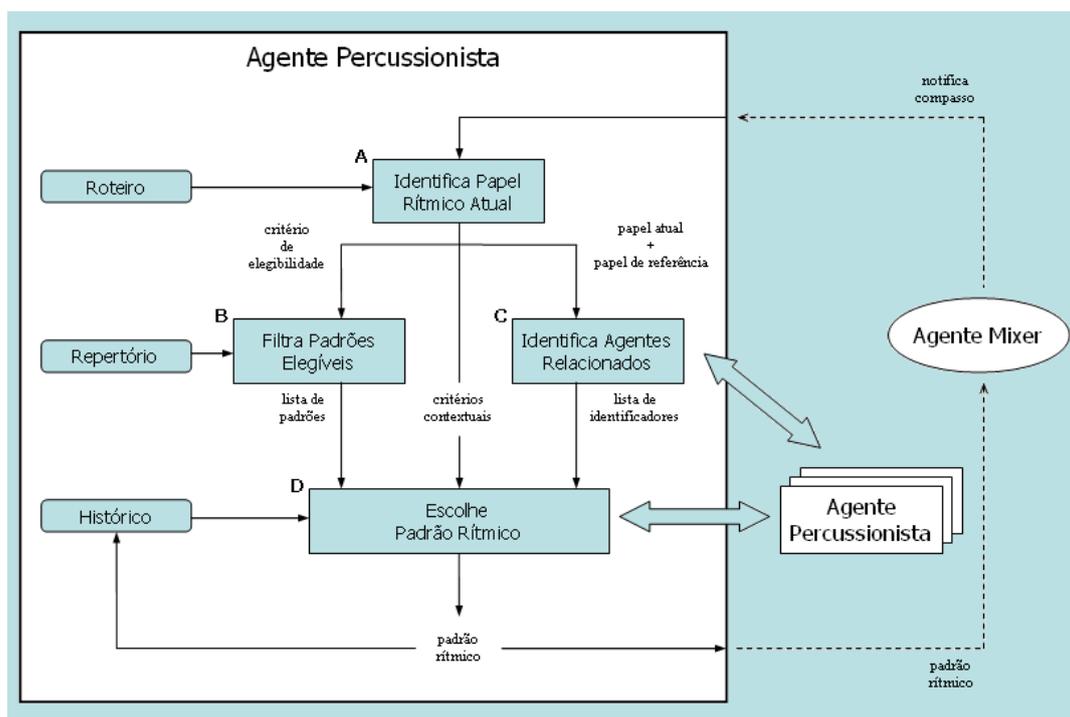
Um **agente percussionista** é criado para cada instrumento adicionado pelo usuário no projeto rítmico. No momento da criação, cada agente recupera de uma **base de padrões rítmicos** do sistema os padrões (monofônicos) do seu instrumento, formando um repertório individual do agente. Assim, o desafio dos agentes é combinar seus respectivos padrões monofônicos em uma performance polifônica interessante. Na figura 5.2, mostramos uma visão geral da arquitetura do sistema CInBalada.



**Figura 5.2:** Arquitetura do sistema CInBalada

A **plataforma multiagente** da figura 5.2 é a plataforma FAMA, apresentada detalhadamente na seção 4.1, enquanto o **agente mixer** é o agente responsável por

gerenciar as saídas sonoras, conforme visto na seção 4.2. O agente *mixer* notifica, periodicamente, a proximidade de ocorrência de um novo compasso da performance. Ao receber essa notificação, cada agente percussionista dá início a um processo de escolha do padrão rítmico. Esse processo envolve **negociação** entre os agentes de modo a garantir a coerência da performance em grupo. Na figura 5.3 mostramos como foi modelado todo o processo de escolha do padrão nos agentes percussionistas do CInBalada, tendo como base o conceito de papéis rítmicos.



**Figura 5.3:** Modelagem interna do agente percussionista

Como pode ser visto na figura 5.3, cada agente percussionista tem três atributos principais:

- **Roteiro:** Sequência de papéis rítmicos que o agente irá assumir a cada compasso. Essa informação vem do projeto rítmico.
- **Repertório:** Lista de padrões rítmicos do instrumento, recuperada da base de padrões rítmicos do sistema.
- **Histórico:** Mantém todos os padrões rítmicos escolhidos pelo agente em compassos anteriores da performance. O histórico é atualizado ao final do processo de escolha do padrão.

Para cada compasso da performance, acontece um novo processo de escolha do padrão rítmico. Esse processo é disparado pela notificação de novo compasso enviada pelo mixer e é dividido nas seguintes etapas (figura 5.3):

- A. **Identifica Papel Rítmico Atual.** Nesta etapa, disparada assim que o agente recebe a notificação de um novo compasso, o agente simplesmente consulta em seu *roteiro* qual o papel rítmico atual. O papel rítmico será usado nas demais etapas para guiar a escolha do padrão rítmico.
- B. **Filtra Padrões Elegíveis.** Usando o *critério de elegibilidade* do papel, o agente filtra os padrões rítmicos do seu repertório. Apenas a lista de padrões rítmicos que satisfazem esse critério é considerada durante o processo de escolha do padrão (etapa D).
- C. **Identifica Agentes Relacionados.** O agente identifica quais agentes percussionistas assumem o mesmo papel rítmico que ele e quais assumem o seu papel rítmico de referência, no dado compasso. Entre esses agentes acontecerá um processo de negociação para escolha do padrão rítmico.
- D. **Escolhe Padrão Rítmico.** Nesta etapa, o agente escolhe o padrão rítmico que irá tocar no compasso atual. Para isso, ele se engaja em um processo de negociação com os agentes relacionados, durante o qual cada agente avalia a adequação do seu padrão rítmico em relação aos dos demais agentes. Essa avaliação é feita por meio de funções de avaliação que representam os três *critérios contextuais* do papel rítmico do agente:
  - *critério de coesão do papel:* usado para comparar o padrão rítmico do agente com os padrões dos agentes de mesmo papel.
  - *critério de adequação à referência:* usado para comparar o padrão do agente com os padrões dos agentes do papel de referência.
  - *critério de adequação à política de evolução:* compara o padrão rítmico com os padrões escolhidos em compassos anteriores (registrados no *histórico*).

Com base nessa avaliação da adequação do padrão rítmico ao contexto da performance, o agente pode decidir efetivar a escolha do padrão ou pode tentar outro padrão de sua lista de padrões. Ao efetivar a escolha, o padrão rítmico é enviado ao agente *mixer* e registrado no histórico.

Nas próximas subseções, apresentamos em mais detalhes os principais componentes da arquitetura do CInBalada. Na subseção 5.4 mostramos a base de padrões rítmicos do sistema. As representações desses padrões usadas em tempo de execução são apresentadas na seção 5.5. O mecanismo de avaliação baseado em papéis rítmicos é descrito em detalhes na seção 5.6. Por fim, o protocolo de negociação usado para escolher os padrões de cada agente é melhor detalhado na seção 5.7.

#### **5.4 Base de Padrões Rítmicos**

O sistema CInBalada mantém uma base de padrões rítmicos monfônicos (i.e. tocados por apenas um instrumento) que são combinados pelos agentes de modo a formar as performances polifônicas do sistema. A base de padrões rítmicos do CInBalada é formada por arquivos MIDI com amostras pequenas (de um a quatro compassos) de performances individuais de instrumentos de percussão. Uma das principais vantagens do formato MIDI é que ele representa de maneira simbólica eventos musicais (como ataques de notas e andamento), facilitando a manipulação dos padrões rítmicos.

Outra vantagem do formato MIDI é que ele permite registrar características de expressividade típicas de performances humanas, como: dinâmica e desvios expressivos. No entanto, para registrar essas características expressivas é necessário que cada padrão seja seqüenciado por um músico humano por meio de uma percussão MIDI. Uma alternativa seria usar um algoritmo para simular as características expressivas humanas, mas este é um problema bastante difícil (Bilmes, 1993).

Ao montar a atual base de padrões do projeto não foi possível seqüenciar os arquivos a partir de performances reais. Em geral, os padrões utilizados foram seqüenciados manualmente no software Cakewalk SONAR (<http://www.cakewalk.com/>) a partir de partituras e, portanto, sem características de expressividade. As partituras foram obtidas de registros de manifestações populares percussivas, como as batucadas de samba (Salazar, 1991) e o maracatu (Santos & Resende, 2005). Também usamos alguns padrões rítmicos encontrados na literatura pesquisada: ritmos clave africanos e afro-americanos (Toussaint, 2002, 2003b) e padrões de palmas no flamenco (Diaz-Báñez et al., 2004).

Apesar de não ter expressividade, o fato de utilizarmos registros (em partituras) de performances humanas deve garantir que padrões que têm qualidade comparável aos padrões produzidos por músicos humanos. Além disso, com essa base de origem

heterogênea, combinações inusitadas entre padrões rítmicos de manifestações musicais distintas podem surgir no CInBalada.

Devido a algumas limitações do formato MIDI (que guarda as informações de performance), o sistema também mantém outras informações de mais alto nível na base, divididas em informações sobre os instrumentos e informações sobre os padrões. Para cada instrumento, o sistema guarda as seguintes informações:

- **Nome do instrumento.** Identifica unicamente o instrumento de percussão.
- **Lista de notas MIDI.** O sistema guarda a lista de possíveis notas MIDI para o dado instrumento. Para cada nota, o cadastro também guarda sua frequência relativa (alta, média ou baixa), para instrumentos que podem produzir diferentes notas (e.g. congas).

Já as informações complementares armazenadas para cada padrão rítmico são:

- **Nome do Instrumento.** Identifica a entrada no cadastro de instrumentos.
- **Nome do arquivo.** Identifica unicamente o arquivo MIDI dentro diretório de padrões rítmicos do sistema.
- **Número da pista (*track*).** Indica em qual pista do arquivo estão armazenadas as informações do padrão<sup>13</sup>.
- **Nome do padrão.** Pode ser usado para identificar a origem do padrão (e.g. “samba-agogo-01”). O sistema não exige que seja único (o padrão é identificado unicamente pelo nome do arquivo).
- **Tipo do padrão.** Indica se é um padrão de virada ou não. Essa informação foi utilizada devido à dificuldade de classificar automaticamente os padrões de virada.
- **Métrica.** Indica a métrica do padrão rítmico. O número de pulsos atômicos (*tatums*) é o único componente da métrica representado. Todos os padrões encontrados são de métricas regulares de 16 pulsos atômicos (geralmente métricas 4/4) ou de 12 pulsos atômicos (12/8, em notação tradicional) ou compatíveis (e.g. métricas de 8 ou 6 pulsos).

---

<sup>13</sup> Apesar dessa informação não ter utilidade na presente versão do sistema, futuramente ela permitirá que mais de um padrão seja guardado em diferentes pistas de um mesmo arquivo.

- **Estilos musicais.** Indica as manifestações musicais nas quais o padrão rítmico é tipicamente encontrado. Um mesmo padrão rítmico de caixa pode estar presente, por exemplo, no frevo e no maracatu.

A base de padrões do sistema pode ser livremente alterada por usuários do sistema, que podem remover ou adicionar padrões e instrumentos. Para isso, ele precisa fornecer todas as informações listadas para o padrão e para o instrumento (caso ele ainda não esteja cadastrado).

Esta seção mostrou as informações mantidas persistentemente na base de padrões. Em tempo de execução, essas informações são convertidas em modelos mais adequados de representação da estrutura rítmica, mostrados na próxima seção.

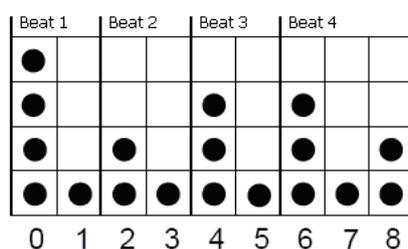
## **5.5 Representação da Estrutura Rítmica**

Na seção 3.2, mostramos modelos de representação encontrados na literatura para os diversos elementos estruturais da dimensão temporal da música. Os modelos ali apresentados são baseados em estudos da percepção musical humana e, portanto, são mais adequados para a criação de composições criativas como pretendemos no CInBalada. Nesta subseção mostramos quais dos modelos analisados foram utilizados no CInBalada, para cada um dos elementos estruturais rítmicos apresentados na seção 3.1.

Quanto à representação da *categoria rítmica*, usamos a abordagem de múltiplas representações paralelas. A representação mais básica do padrão é o formato MIDI, composto de informações de performance a partir das quais as outras representações são calculadas. Primeiramente, o sistema quantiza o padrão segundo o pulso atômico subjacente (cadastrado como informação complementar na base). Desse modo, obtém-se a representação **TUBS** do padrão rítmico. A partir da representação TUBS, o sistema calcula o **vetor de posições de ataques** e, a partir deste, o **vetor de intervalos entre ataques**, conforme explicado na seção 3.2. A partir do vetor de intervalos entre ataques, o sistema calcula, por fim, o **vetor cronotônico**. Essas representações do padrão rítmico foram escolhidas por permitirem o cálculo das medidas que o sistema usa como funções de avaliação dos padrões rítmicos (vide próxima seção).

Já para a métrica, o sistema usa uma representação derivada da **estrutura métrica** apresentada na seção 3.2. A diferença é que o sistema suporta métricas complexas na forma de justaposição de métricas simples. Por exemplo, a métrica quaternária complexa mostrada na figura 3.14 pode ser representada no sistema pela estrutura métrica da figura

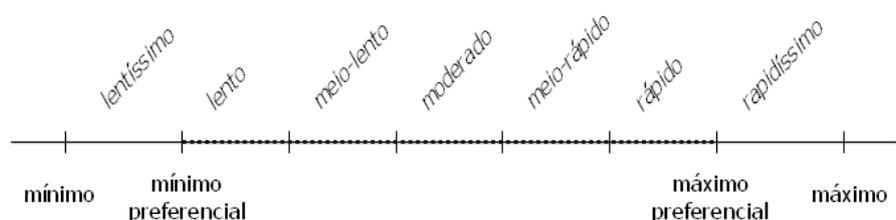
5.4, que é apenas a justaposição de uma métrica regular de três batidas de dois tatums com uma métrica de uma batida de três tatums.



**Figura 5.4:** Representação de uma métrica quaternária complexa no CInBalada

O CInBalada representa o andamento do sistema usando a unidade BPM (semínimas por minuto), devido à difusão desta medida (ela é usada no formato MIDI, por exemplo). O sistema estabelece limites máximo e mínimo sobre os valores possíveis do andamento com base nas restrições perceptuais para duração dos pulsos apresentadas na seção 3.1. O sistema usa o valor mínimo de duração do pulso atômico para calcular o **andamento máximo** permitido e usa o valor máximo da soma dos pulsos da métrica para calcular o **andamento mínimo** permitido. Situados entre esses dois valores, o sistema calcula dois outros limites chamados “preferenciais” por serem calculados com base em restrições perceptuais adicionais. O **andamento preferencial mínimo** é calculado a partir da duração preferencial máxima para pulsos maiores, enquanto o **andamento preferencial máximo** é calculado a partir da duração preferencial mínima para pulsos maiores (apresentados na seção 3.1).

O sistema usa esses quatro limites para definir gradações do andamento e atribuir-lhes valores qualitativos baseados em restrições perceptuais (vide figura 5.5). Com isso, o sistema consegue representar de forma mais intuitiva o andamento por meio de **adjetivos de andamento**, como acontece com a notação tradicional. Porém, o sistema apresenta a vantagem sobre a notação tradicional de criar maior distinção entre os adjetivos de andamento, pois não há sobreposição entre eles.



**Figura 5.5:** Andamento no CInBalada

Após mostrar como os padrões são armazenados na base do sistema e como são representados em tempo de execução, apresentamos, na próxima seção, como os agentes do CInBalada fazem a avaliação dos padrões rítmicos usando essas representações.

## 5.6 Mecanismo de Avaliação

Para que o CInBalada gere saídas musicais relevantes (i.e. de boa qualidade musical), é importante que cada agente escolha adequadamente os padrões rítmicos que serão tocados em conjunto. Para isso, é necessário que os agentes tenham um mecanismo de avaliação que julgue a validade de cada padrão em cada possível contexto. Já explicamos no final da seção 5.2 que o conceito de *papéis rítmicos* pode ser útil para a implementação de um mecanismo de avaliação em um sistema de composição rítmica. Nesta seção, mostraremos como alguns dos papéis rítmicos apresentados naquela seção foram modelados no CInBalada.

Inicialmente vamos mostrar como cada um dos cinco elementos dos papéis rítmicos estão presentes na arquitetura do CInBalada:

- a. **Critério de elegibilidade para padrões rítmicos individuais.** Funciona como uma função de avaliação para padrões individuais. Quando o valor da função está dentro de um certo limiar, o padrão é considerado elegível para exercer o papel rítmico; caso contrário, o padrão é descartado. Usado na etapa “Filtrar Padrões Elegíveis” da figura 5.3.
- b. **Papel de referência.** Representado simplesmente pelo nome do papel.
- c. **Critério de adequação ao papel de referência.** Está presente no sistema na forma de uma função de avaliação da combinação de um padrão do papel e um padrão do papel de referência.
- d. **Critério de coesão do papel.** Também está presente no sistema na forma de uma função de avaliação, porém é aplicada a pares de padrões do mesmo papel para garantir a coesão. Foi implementada, para cada papel, a partir das medidas da seção 3.3 e de medidas novas aqui propostas.
- e. **Política de evolução dos padrões.** Representada por mais uma função de avaliação, que recebe o histórico de padrões escolhidos anteriormente e um padrão para avaliar a adequação deste padrão à política de evolução do papel.

Portanto, o sistema representa os papéis rítmicos, basicamente, por meio de quatro funções de avaliação que são aplicadas para definir qual padrão rítmico é a melhor escolha em um dado compasso. Às três últimas funções chamaremos de *funções de avaliação contextuais*, pois elas modelam os critérios contextuais dos papéis.

As funções de avaliação dos papéis rítmicos do CInBalada foram concebidas com base nas medidas de propriedades rítmicas apresentadas na seção 3.3 e também em medidas novas propostas no presente trabalho. A subseção a seguir apresenta as novas medidas propostas, enquanto na subseção seguinte mostramos como cada papel rítmico teve suas quatro funções de avaliação modeladas com base nas diversas medidas (novas ou não).

### 5.6.1 Novas Medidas de Propriedades Rítmicas

Uma nova medida, que foi definida para ser usada como critério de elegibilidade mínimo em todos os papéis rítmicos do CInBalada, foi o **teste de compatibilidade métrica**. Esse teste consiste simplesmente em verificar se o número de pulsos atômicos da métrica corrente é igual ou múltiplo do número de pulsos atômicos da métrica do padrão rítmico<sup>14</sup>. Esse teste garante que o padrão rítmico vai preencher perfeitamente um compasso da métrica corrente, mesmo que seja necessário incluir um certo número de repetições inteiras dele.

Uma medida bastante simples proposta serve para diferenciar padrões mais ativos dos menos ativos e, portanto, é chamada de **medida de atividade**. Essa medida de atividade consiste simplesmente na divisão do número de ataques pelo número de compassos do padrão rítmico. Apesar de bastante simples, essa foi uma importante medida usada nos *critérios de elegibilidade* dos papéis rítmicos do CInBalada.

Outro critério proposto neste trabalho é uma **medida de interplay** entre dois padrões rítmicos<sup>15</sup>. Ela é calculada a partir da representação TUBS dos padrões. O seu cálculo é baseado na contagem dos dois seguintes valores:

- **encontros**: o número de posições (pulsos atômicos) correspondentes que são ataques em ambos os padrões;

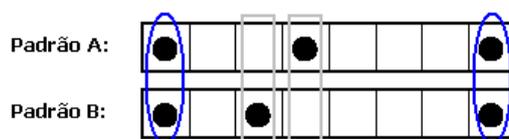
---

<sup>14</sup> A métrica atual é mantida no agente mixer, conforme apresentado na seção 4.4, enquanto o número de pulsos atômicos de um padrão é uma informação cadastrada na base, conforme seção 4.5.

<sup>15</sup> Sugerida por Geber Ramalho.

- **desencontros:** o número de posições correspondentes que são ataques em apenas um dos padrões.

O resultado da função é a diferença absoluta entre esses dois valores. A figura 5.6 destaca os pontos de encontros e desencontros para dois padrões de oito *tatums*. São 2 encontros e 2 desencontros, o que faz a medida assumir o valor zero.



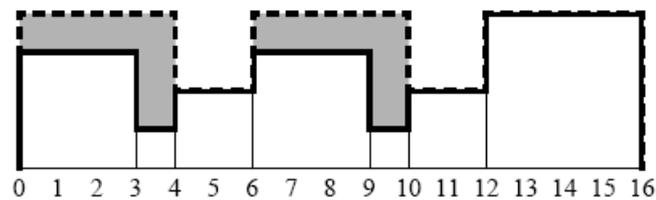
**Figura 5.6:** Medida de *interplay* – encontros e desencontros entre ataques

Quanto menor o valor da medida de *interplay*, maior equilíbrio há entre números de encontros e desencontros e, portanto, melhor é considerado a combinação dos dois padrões. Essa medida baseia-se na idéia de que são necessárias tanto uma certa dose de afinidade (os encontros) entre dois padrões monofônicos, quanto um pouco de originalidade (desencontros) entre eles, para que suas qualidades se somem num padrão polifônico interessante.

A medida de *interplay* era a única função de avaliação presente na primeira versão do sistema (vide seção 5.1), antes que fossem consideradas as medidas encontradas na literatura. Ela foi sugerida quando a proposta do sistema ainda estava sendo validada. Apesar de não ter tido a sua relevância musicológica comprovada, ela continuou a ser usada no sistema diante da falta de uma medida de objetivo semelhante na literatura.

Por fim, a última medida proposta no presente trabalho é uma **medida de derivação** de um padrão em relação a outro tomado como referência. Definimos aqui que um padrão rítmico *i* é *derivado* de um padrão de referência *j* quando todo ataque de *i* também é ataque de *j*. Porém, definida assim, a derivação é simplesmente uma propriedade que pode estar presente ou não. Uma medida de derivação foi criada para lidar com casos intermédios em que um padrão tem apenas algumas partes (e.g. metade do compasso) que podem ser consideradas derivadas de um outro padrão de referência. Essa medida de derivação foi inspirada na medida de *distância cronotônica*.

Mostramos na figura 5.7 a representação TEDAS<sup>16</sup> do ritmo clave *Shiko* e um padrão rítmico derivado dele.



**Figura 5.7:** Padrão Shiko (tracejado) e padrão derivado (traço contínuo)

De maneira simplificada, o cálculo da medida de derivação consiste na soma das diferenças entre o padrão tomado como referência e o suposto padrão derivado<sup>17</sup>. Esse cálculo é quase igual ao da distância cronotônica se não fosse por uma diferença fundamental: as diferenças reais é que são somadas, ao invés das diferenças absolutas usadas na distância cronotônica. Assim, um padrão rítmico será considerado (parcialmente) derivado de um dado padrão de referência quando a soma das diferenças entre os vetores cronotônicos for positiva. Se a soma for negativa, a relação de derivação é invertida, ou seja, o suposto padrão de referência é derivado do primeiro padrão. Se a soma for zero, os padrões são iguais ou não são derivados um do outro.

Se um padrão for completamente derivado do padrão de referência (como na figura 5.7), o valor do cálculo apresentado acima será igual ao valor da distância cronotônica entre os padrões. Na verdade, a distância cronotônica é o valor máximo que a medida apresentada pode apresentar. Por isso, no CinBalada, o valor final da medida de derivação é, também, dividido pela distância cronotônica. Assim, a medida de derivação proposta retorna os seguintes valores quando aplicada a um par de padrões ( $a$ ,  $r$ ):

- **+1**: o padrão  $a$  é completamente derivado do padrão de referência  $r$
- **(0; +1)**: o padrão  $a$  é parcialmente derivado do padrão de referência  $r$
- **0**: os padrões não são derivados
- **(0; -1)**: o padrão de referência  $r$  é parcialmente derivado do padrão  $a$ ;
- **-1**: o padrão  $r$  é completamente derivado do padrão  $a$

<sup>16</sup> Esta representação é análoga ao vetor cronotônico, conforme visto na seção 3.2.

<sup>17</sup> Diferente de outras medidas apresentadas neste trabalho, a *medida de derivação* leva em consideração a ordem, ou seja, se ela for aplicada ao par de padrões ( $a,r$ ) ela resultará em um valor distinto (simétrico) de sua aplicação ao par ( $r,a$ ).

Essa medida foi proposta para ser usada como critério de adequação ao papel base (quando ele serve de papel de referência), pois o padrão tocado por este papel serve para organizar a performance, tocando os pulsos da métrica ou um ritmo clave (conforme definição do papel *base* na seção 5.2).

A seguir mostramos como as novas medidas bem como as medidas definidas na seção 3.3 serviram para modelar os papéis rítmicos no CInBalada.

### 5.6.2 Modelagem dos Papéis Rítmicos

Como visto no início da seção, os papéis rítmicos do CInBalada foram implementados por meio de quatro funções de avaliação, que representam os quatro elementos dos papéis rítmicos definidos na seção 5.2. Para compor cada função, foram escolhidas algumas das medidas de propriedades rítmicas apresentadas neste trabalho. A seguir, veremos a escolha feita para cada um dos papéis rítmicos incluídos no CInBalada.

As funções de avaliação que compõem o papel *base* do CInBalada foram modeladas da seguinte maneira:

- a. **Critério de elegibilidade:** usa as medidas de *atividade* e a medida de *igualdade de espaçamento* por regressão linear. Consoante com a definição do papel na seção 5.2, são considerados elegíveis os padrões de baixa atividade e grande igualdade de espaçamento.
- b. **Critério de adequação à referência:** não possui referência.
- c. **Critério de coesão:** usa a *distância cronotônica*, para garantir que todos os instrumentos do papel toquem padrões semelhantes. Essa medida foi escolhida para evitar que a performance ficasse confusa na presença de vários instrumentos base.
- d. **Política de evolução:** usa a *distância cronotônica* para comparar um padrão com os últimos quatro padrões escolhidos para o mesmo papel. Quanto mais similar um dado padrão for dos quatro últimos, melhor. Isso garante que o papel não permita variação nos seus padrões (conforme definição do papel *base* na seção 5.2).

O papel *base complementar* foi modelado da seguinte maneira:

- a. **Critério de elegibilidade:** como o papel *base*, usa as medidas de *atividade* e a medida de *igualdade de espaçamento* por regressão linear. Porém, diferentemente do papel *base*, são considerados elegíveis os padrões de alta atividade e razoável igualdade de espaçamento.

- b. **Critério de adequação à referência:** A referência é o papel *base*. Usa a *medida de derivação* para garantir que os padrões são derivados (parcialmente, pelo menos) dos padrões executados por instrumentos do papel *base*.
- c. **Critério de coesão:** usa a *medida de interplay* para tentar tornar os interessante a combinação dos padrões tocados por instrumentos desse papel.
- d. **Política de evolução:** usa a *distância cronotônica* para comparar um padrão com os últimos dois padrões escolhidos para o mesmo papel. Quanto mais similar um dado padrão for dos dois últimos, melhor. Assim, o papel *base complementar* permite um pouco mais de variação do que o papel *base*.

Já o papel *solo* foi modelado da seguinte maneira:

- a. **Critério de elegibilidade:** também usa as medidas de *atividade* e a medida de *igualdade de espaçamento* por regressão linear e ainda a medida de *offbeatness*. São considerados elegíveis os padrões de alta atividade e baixa igualdade de espaçamento (conforme análise desse papel na seção 5.2), além de preferir padrões com alto valor de *offbeatness*, para garantir que os padrões são interessantes individualmente<sup>18</sup>.
- b. **Critério de adequação à referência:** assim como o papel *base complementar*, o papel *solo* tem o papel *base* como referência e usa a *medida de derivação* para garantir que seus padrões são derivados dos padrões dos instrumentos *base*.
- c. **Critério de coesão:** como no papel *base complementar*, usa a *medida de interplay* para tentar tornar os padrões do papel equilibrados entre si.
- d. **Política de evolução:** usa a *distância cronotônica* para garantir que o padrão não será igual aos dois últimos padrões escolhidos para o papel *solo*. Desse modo tentamos garantir que o papel *solo* estaria escolhendo padrões diferentes a cada compasso.

Por fim, o papel *virada* foi modelado conforme descrito abaixo:

---

<sup>18</sup> A seção 3.3 mostra que a medida de *offbeatness* pode servir como uma medida de preferência para padrões individuais.

- a. **Critério de elegibilidade:** apesar de serem fáceis de serem identificados por ouvintes humanos, os padrões de virada são difíceis de serem descritos com as medidas propostas. Por esse motivo, padrões de virada já vêm classificados da base pelo campo “tipo do padrão” (vide seção 5.4).
- b. **Critério de adequação à referência:** não possui referência.
- c. **Critério de coesão:** usa a *distância cronotônica* para garantir que viradas executadas simultaneamente por vários instrumentos sejam similares (caso contrário, as viradas soariam caóticas em conjunto).
- d. **Política de evolução:** livre, ou seja, o histórico não é levado em consideração na escolha do padrão.

A divisão dos papéis rítmicos em quatro funções de objetivos claros e distintos facilitou o trabalho de modelagem de cada função. Porém, nos casos em que a função foi modelada com mais de um critério, foi difícil balancear os pesos relativos (e também o limiar, nos critérios de elegibilidade). Por isso, esse trabalho de ajuste dos pesos acabou sendo feito empiricamente e apenas para padrões da métrica quaternária (16 pulsos). Para as métricas ternárias, seria necessário rever todos os pesos e limiares, até mesmo devido às características de algumas medidas usadas, que são dependentes do número de tatum.

A política de evolução do papel *solo* também merecia ser analisada com mais cuidado. A ligação entre os padrões consecutivos precisa ser considerada de alguma maneira. Infelizmente, não foi possível encontrar (ou criar) alguma medida que tratasse dessa continuidade dos padrões escolhidos em compassos consecutivos. A tabela 5.1 a seguir resume a modelagem dos quatro papéis rítmicos presentes no CInBalada.

	<i>critério de elegibilidade</i>	<i>critério de adequação</i>	<i>critério de coesão</i>	<i>política de evolução</i>
<b>Base</b>	baixa atividade alta regularidade	(não tem papel de referência)	similaridade	similar aos quatro últimos
<b>base complementar</b>	alta atividade alta regularidade	derivação em relação ao papel base	interplay	similar aos dois últimos
<b>Solo</b>	alta atividade baixa regularidade alto offbeatness	derivação em relação ao papel base	interplay	diferente dos dois últimos
<b>Virada</b>	(classificação carregada da base)	(não tem papel de referência)	similaridade	livre

**Tabela 5.1:** Modelagem dos papéis rítmicos do CInBalada

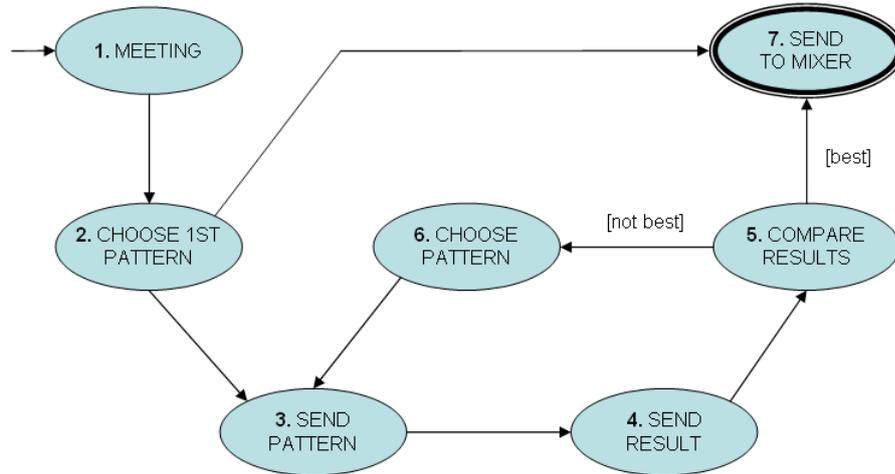
Das medidas analisadas na seção 3.3, o sistema utiliza: a distância cronotônica (para similaridade), a medida de *offbeatness* e a medida de igualdade de espaçamento por regressão linear (para regularidade). Além disso, o sistema usa as medidas aqui propostas: teste de compatibilidade métrica, a medida de atividade, a medida de derivação e a medida de *interplay*.

Na verdade, toda a modelagem proposta nesta subseção consiste em uma alternativa de modelagem para os papéis rítmicos. A partir das mesmas definições apresentadas na seção 4.2, outras possíveis modelagens também podem ser propostas, possivelmente usando outras medidas de avaliação não analisadas no presente trabalho.

### **5.7 Negociação entre Agentes Percussionistas**

Os papéis rítmicos definem certas relações de dependência entre os instrumentos, para garantir uma certa coerência global na performance. As relações existem em dois níveis: do instrumento com instrumentos do seu papel referência (relacionamento vertical) e do instrumento com instrumentos do mesmo papel (relacionamento horizontal). Assim, o padrão rítmico escolhido por um agente influencia os padrões que podem ser escolhidos por outros agentes. Essas dependências mútuas entre os agentes são resolvidas por meio do protocolo de negociação do CInBalada.

Assim, os *objetos de negociação* (Jennings, et al., 2001) são os padrões rítmicos a serem tocados por cada cada agente, ou seja, a negociação visa selecionar o padrão que cada agente irá tocar. Para isso, cada agente usa como *modelo de decisão* as três funções de avaliação contextuais do papel rítmico selecionado. O *protocolo de negociação* do CInBalada foi modelado com a máquina de estados exibida na figura 5.8. Essa máquina de estados engloba tanto o passo “Identifica Agentes Relacionados” como o passo “Escolhe Padrão Rítmico” da figura 5.3 (pois ambos envolvem interação entre os agentes).



**Figura 5.8:** Máquina de estados do protocolo de negociação

O **estado 1** (*meeting*) é ponto de entrada da máquina de estados, acionado no momento em que o agente mixer envia uma notificação de compasso. Nesse estado, os agentes se encontram usando o serviço de páginas amarelas da plataforma FAMA e informam uns aos outros quais seus respectivos papéis rítmicos naquele compasso. De acordo com os papéis rítmicos, cada agente encontra os *agentes relacionados* a ele, ou seja: agentes do mesmo papel ou agentes de referência para ele (ou, ainda, agentes para os quais ele serve de referência). Este estado equivale à atividade “Identifica Agentes Relacionados” da figura 5.3, enquanto os demais, juntos, acontecem na atividade “Escolhe Padrão Rítmico” da figura 5.3.

No **estado 2** (*choose 1st pattern*), os agentes fazem a primeira *escolha parcial* (não enviada ao *mixer*) de um padrão rítmico para o compasso atual. Antes de fazer tal escolha, o agente consulta no histórico o último padrão efetivamente escolhido (ou seja, enviado ao *mixer*) para o mesmo papel. Se existir, a lista de padrões (repertório) do agente será ordenada por similaridade em relação a esse padrão; senão, o agente irá ordenar sua lista de padrões aleatoriamente. O padrão escolhido será, então, o primeiro padrão da lista. O restante do processo de negociação se dará apenas entre agentes relacionados. Se o agente não tiver agentes relacionados, ele vai direto para o estado 7.

No **estado 3** (*send pattern*) os agentes relacionados simplesmente enviam uns aos outros os padrões escolhidos no estado anterior. No **estado 4** (*send result*), já em posse dos padrões dos agentes relacionados, cada agente aplica as três funções de avaliação contextuais do papel para cada padrão elegível:

- A função que representa o *critério de adequação à referência* é aplicada para todos os padrões vindos de agentes do papel de referência.
- A função que representa o *critério de coesão* é aplicado, para todos os padrões vindos de agentes do mesmo papel.
- A função que representa a *política de evolução* é aplicada usando o histórico do agente.

Ainda no estado 4, o resultado combinado das três funções é enviado aos agentes de mesmo papel.

No **estado 5** (*compare results*), cada agente compara o seu próprio resultado da avaliação com os resultados enviados pelos seus pares. O agente de melhor valor de avaliação de cada papel é identificado (inclusive os empatados) e passa para o **estado 7** (*send to mixer*), onde o padrão é *efetivado* sendo enviado ao agente *mixer* e inserido no histórico. Os demais agentes seguem para o estado 6 para que cada um escolha outro padrão.

No **estado 6** (*choose pattern*), cada agente define, para a sua lista de padrões, uma *janela* de 20 posições vizinhas da última escolha parcial. O agente percorre essa janela aplicando as três funções de avaliação contextuais e calculando o resultado combinado delas (do mesmo modo que acontece no estado 4) usando os últimos padrões enviados pelos agentes relacionados. Padrões fora da janela não são considerados. Após a escolha do melhor padrão da janela, o agente volta ao estado 3.

Cada agente, portanto, fica repetindo o ciclo de estados 3-4-5-6 até que o seu padrão tenha o melhor resultado da rodada. Esse ciclo garantidamente termina para todos os agentes pelo fato de que, a cada rodada, um agente (ou mais, se houver empates) sai do ciclo em questão para efetivar o padrão escolhido.

### 5.7.2 Comentários Sobre o Protocolo

No estado 2, no caso em que *não há* entrada no histórico, mostramos que é realizada uma ordenação aleatória do repertório, sendo escolhido o padrão que ficar na primeira posição. A justificativa para essa escolha aleatória é que, naquele ponto da negociação, nenhum agente escolheu seu padrão ainda e, portanto, não há contexto algum para aplicar as funções de avaliação contextuais do papel. Outra motivação para a escolha aleatória é gerar diversidade (que é um dos requisitos do sistema), pois ela faz com que a cada nova performance o primeiro padrão escolhido seja diferente.

No mesmo estado, no caso em que *há* entrada no histórico, mostramos que o repertório é ordenado por similaridade em relação à última escolha. Isso acontece porque, no estado 6, o agente vai fazer uma escolha cega, sem conhecimento da escolha que os outros agentes estão fazendo em paralelo. A ordenação por similaridade, em conjunto com a janela definida no estado 6, garante que os padrões que serão escolhidos pelos outros agentes serão razoavelmente similares aos últimos que foram enviados (no estado 3). Além disso, à medida que os agentes vão efetivando suas escolhas (estado 7), mais fácil se torna a escolha para os agentes em negociação, pois os padrões daqueles agentes não serão mais alterados.

Sobre a janela definida no estado 6, ela serve para reduzir o espaço de busca e, conseqüentemente, tornar o sistema mais escalável para mais padrões rítmicos e para mais agentes. Em conjunto com a ordenação aleatória do estado 2, a janela garante ao sistema a geração de uma certa diversidade, pois limita os padrões atingíveis a um subconjunto aleatório. Isso faz com que execuções distintas de um mesmo projeto rítmico do CInBalada possam gerar composições distintas entre si.

## **5.8 Considerações Finais**

Este capítulo apresentou a modelagem do sistema CInBalada, cujo propósito é servir como um sistema de composição rítmica que satisfaz os requisitos de *qualidade e diversidade*. Agentes percussionistas do sistema são responsáveis por criar performances rítmicas polifônicas por meio da combinação de padrões rítmicos monofônicos pré-gravados, obtidos de diversas manifestações rítmicas humanas.

O sistema foi proposto com base no conceito de *papel rítmico*, introduzido para tentar dar ao usuário do sistema controle musical de alto nível sobre as saídas do sistema. Esse conceito guiou a elaboração de um mecanismo de avaliação multicritério, usado por cada agente para escolher um padrão rítmico adequado ao contexto da performance. A escolha dos padrões rítmicos dos agentes acontece por meio de um protocolo de negociação, usando o mecanismo de avaliação baseado em papéis rítmicos como modelo de decisão dos agentes.

A modelagem proposta oferece aos usuários do sistema os seguintes parâmetros de configuração das composições resultantes do sistema: alteração da orquestração, livre distribuição dos papéis rítmicos entre os instrumentos, configuração da base (adição e remoção de padrões), além de parâmetros básicos de configuração da performance (oferecidos por meio de agente *mixer*). Os resultados podem ser ouvidos pelo usuário em

tempo real, permitindo que o usuário faça experimentações com arranjos rítmicos facilmente.

## CAPÍTULO 6

# Implementação do CInBalada

No capítulo anterior, mostramos a modelagem do sistema CInBalada. Neste capítulo, apresentamos como aquela modelagem foi implementada. A linguagem escolhida para implementação foi C++ (Stroup, 1997) devido, primeiramente, às suas características de orientação a objeto que trazem todas as vantagens deste paradigma (e.g. encapsulamento, modularidade e reusabilidade). Em segundo lugar, devido ao seu alto desempenho de execução, que é um requisito de grande importância em aplicações musicais como o CInBalada (assim como em aplicações multimídias de um modo geral). O ambiente de desenvolvimento usado foi o Borland C++ Builder 6<sup>19</sup> e o sistema usou as bibliotecas TinyXml<sup>20</sup>, para manipulação de XML, e Maximum MIDI (Messick, 1998), para manipulação de arquivos e saídas MIDI.

Na primeira seção deste capítulo, descrevemos como o sistema implementa os mecanismos de armazenamento e manipulação dos padrões rítmicos do sistema. A segunda seção aborda a implementação dos papéis rítmicos como um mecanismo de avaliação. Já na terceira seção, apresentamos como os agentes percussionistas do sistema foram implementados a partir do FAMA (cuja implementação já foi apresentada na seção 4.1.2). Por fim, na quarta seção, mostramos como acontece a integração das diversas partes do sistema por meio de uma fachada.

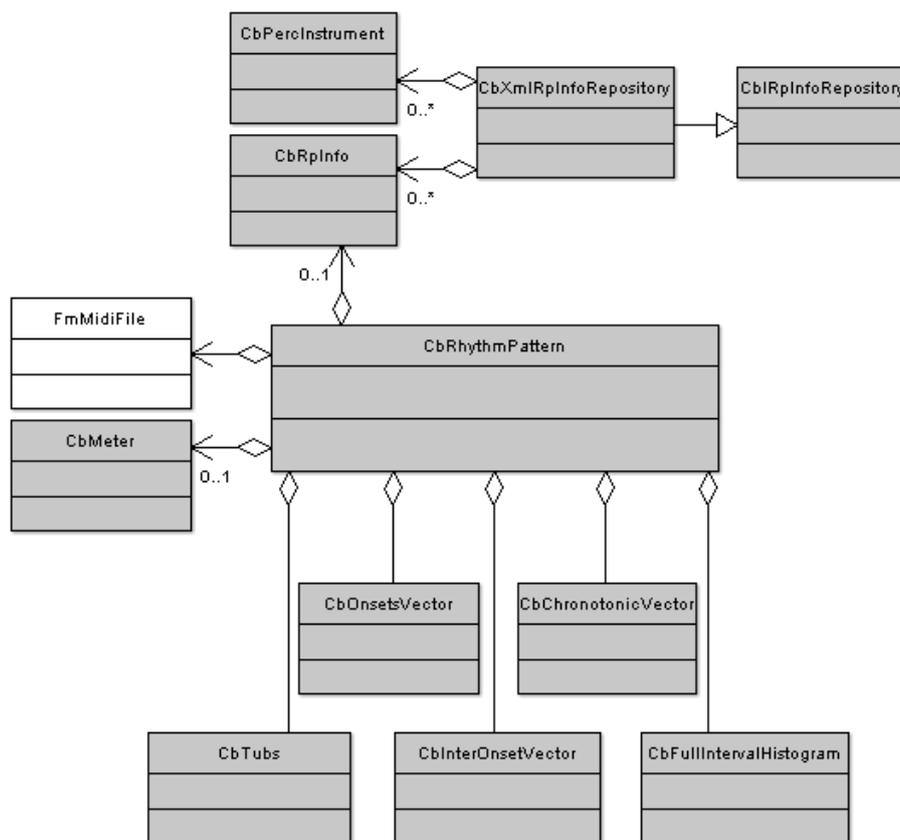
### **6.1 Padrões Rítmicos**

A manipulação de padrões rítmicos monofônicos pré-existent é parte central do sistema CInBalada. Um conjunto de classes do sistema é responsável por essa manipulação. Essas classes e seus relacionamentos são mostrados no diagrama da figura 6.1 (em branco estão representadas as classes do FAMA, enquanto as classes mais escuras são específicas do CInBalada).

---

<sup>19</sup> <http://www.borland.com/cbuilder>

<sup>20</sup> <http://sourceforge.net/projects/tinyxml>



**Figura 6.1:** Classes que representam e armazenam os padrões rítmicos

Os padrões rítmicos do sistema são armazenados como arquivos MIDI individuais (representados em tempo de execução pela classe `FmMidiFile`, do FAMA). Para cada um desses arquivos, um conjunto de informações de alto nível (vide seção 5.4) é mantido persistentemente e carregado em instâncias da classe `CbRpInfo`<sup>21</sup> quando o sistema é carregado. Também são mantidas, persistentemente, informações sobre os instrumentos de percussão disponíveis no sistema (seção 5.4). Estas informações são carregadas em tempo de execução pela classe `CbPercInstrument`.

As informações de alto nível sobre os padrões e instrumentos são cadastradas em um repositório representado pela interface `CbIRpInfoRepository`. O sistema possui apenas uma implementação dessa interface – a classe `CbXmlRpInfoRepository`, que guarda todas as citadas informações em um arquivo XML central. Portanto, para estender a base do `CInBalada`, basta fornecer à classe `CbXmlRpInfoRepository` o arquivo

<sup>21</sup> O prefixo “Cb” é o padrão para classes do `CInBalada`, enquanto “Rp” é uma abreviação de *rhythm pattern* (padrão rítmico) usada para evitar nomes de classes muito longos.

MIDI do padrão rítmico e as informações complementares mostradas na seção 5.4. O arquivo MIDI é, então, copiado para o diretório de padrões do sistema e as informações complementares são salvas em um arquivo XML único para toda base. Na figura 6.2 mostramos um exemplo de entrada XML para um padrão rítmico e um instrumento.

```
<PercussionInstrument name="agogo">
  <Note midinote="67" relpitch="high" />
  <Note midinote="68" relpitch="low" />
</PercussionInstrument>
<RhythmPattern name="bs-agogo-02" instrument="agogo"
  filepath="bs-agogo-02.mid" track="1">
  <MusicalStyle name="samba" />
  <MusicalStyle name="rumba" />
  <Meter tatums="16" />
```

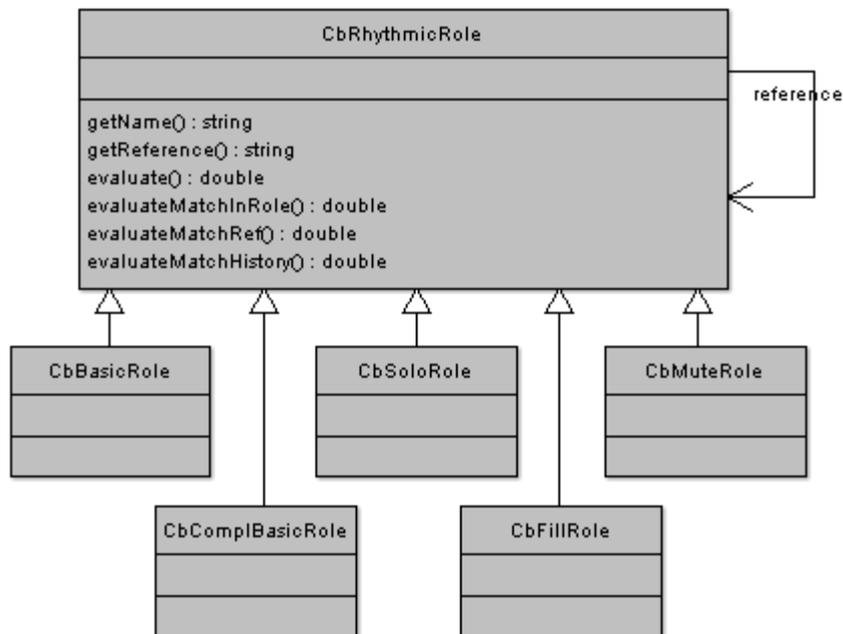
Figura 6.2: Entrada XML para um padrão rítmico e um instrumento

A partir de uma instância de `CbRpInfo`, é possível criar uma instância da classe `CbRhythmPattern`, que representa todo o conhecimento disponível no `CInBalada` sobre um padrão rítmico. Nessa classe, são carregadas as informações de performance gravadas no arquivo MIDI do padrão por meio da classe `FmMidiFile`, bem como a representação da métrica do padrão (mostrada na seção 5.5) por meio da classe `CbMeter`. Partindo dessas duas informações, são construídas as diversas representações alternativas do padrão por meio das classes `CbTubs` (representação TUBS), `CbOnsetsVector` (vetor de ataques), `CbInterOnsetVector` (vetor de intervalos entre ataques), `CbChronotonicVector` (vetor cronotônico) e `CbFullIntervalHistogram` (histograma completo de intervalos). Na classe `CbRhythmPattern` também são implementadas todas as diversas medidas básicas de avaliação disponíveis no sistema (e.g. atividade, igualdade de espaçamento, *offbeatness* e distância cronotônica) fazendo uso das diversas representações alternativas.

## 6.2 Papéis Rítmicos

Os papéis rítmicos são implementados no `CInBalada` pela classe abstrata `CbRhythmicRole`. Como pode ser visto no diagrama de classes da figura 6.3, as seguintes classes herdam de `CbRhythmicRole`, representando os diversos papéis rítmicos disponíveis no sistema: `CbBasicRole` (papel base), `CbComplBasicRole`

(papel base complementar), `CbSoloRole` (papel solo) e `CbFillRole` (papel virada). A classe `CbMuteRole`, apesar de não ser um papel rítmico previsto na seção 5.6, representa a ausência de ataques e é usada para permitir que um instrumento seja configurado para permanecer mudo em um compasso.



**Figura 6.3:** Classes que representam os papéis rítmicos

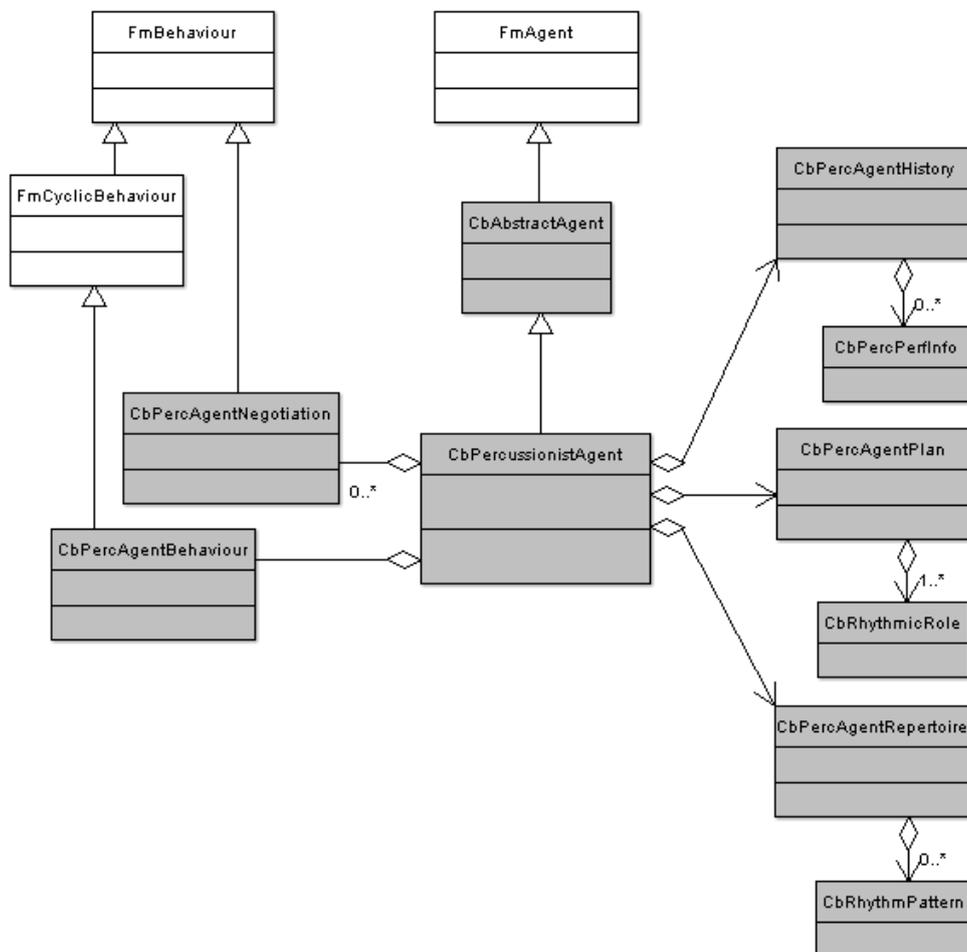
A classe abstrata `CbRhythmicRole` define as operações que as classes filhas devem implementar de modo a representar o papel rítmico como um mecanismo de avaliação (conforme seção 5.6):

- `getName()`: não é um elemento previsto na seção 5.2, mas é necessário para identificar unicamente o papel rítmico.
- `getReference()`: nome do papel de referência.
- `evaluate()`: avalia um padrão rítmico pelas características individuais que apresenta, segundo o *critério de elegibilidade* do papel.
- `evaluateMatchInRole()`: avalia a qualidade da combinação de dois padrões do papel rítmico, conforme o *critério de coesão* do papel.
- `evaluateMatchRef()`: avaliar a qualidade da combinação de um padrão rítmico do papel com um padrão do papel de referência, conforme o *critério de adequação à referência*.

- `evaluateMatchHistory()`: avalia o grau de satisfação da *política de evolução* do papel rítmico, dado um padrão rítmico do papel e o histórico de padrões tocados anteriormente.

### 6.3 Agentes Percussionistas

O agente percussionista baseado em papéis rítmicos é implementado por um conjunto de classes, que são centralizadas pela classe `CbPercussionistAgent`, como pode ser visto na figura 6.4 (em branco estão representadas as classes do FAMA).



**Figura 6.4:** Classes que compõem o agente percussionista

O agente `CbPercussionistAgent` não herda diretamente da classe `FmAgent` (do FAMA), mas de uma classe abstrata chamada `CbAbstractAgent`, que oferece algumas operações gerais necessárias para implementar um agente percussionista. Essas operações são independentes da técnica utilizada para escolher os padrões rítmicos e, assim, permitiu que a classe `CbAbstractAgent` fosse usada durante o desenvolvimento

para implementar diferentes versões de agentes percussionistas (e.g. um agente que escolhe os padrões aleatoriamente).

O agente percussionista baseado em papéis possui dois *behaviours*: o *CbPercAgentBehaviour* e o *CbPercAgentNegotiation*. O primeiro, executado ciclicamente, é responsável por responder eventuais mensagens de requisição do papel rítmico ou do padrão (parcialmente) escolhido do agente, para um dado compasso. O primeiro *behaviour* também é responsável por criar instâncias do segundo *behaviour* a cada nova notificação de compasso enviada pelo agente *mixer*. Este segundo *behaviour* implementa todo o processo de negociação detalhado na seção 5.7.

Os atributos de um agente percussionista apresentados na seção 5.3 são implementados por instâncias das classes: *CbPercAgentReperoire* (repertório de padrões do agente), *CbPercAgentPlan* (lista de papéis rítmicos que serão assumidos a cada compasso) e *CbPercAgentHistory* (informações sobre as escolhas feitas a cada compasso no decorrer da performance).

## 6.4 Fachada

As diversas partes do sistema mostradas nas seções anteriores são acessíveis por meio da classe *CbFacade*, que serve como ponto central de acesso às funcionalidades do sistema *CInBalada*. Na figura 6.5 mostramos a classe e seus interrelacionamentos (novamente, as classes em branco fazem parte do FAMA).

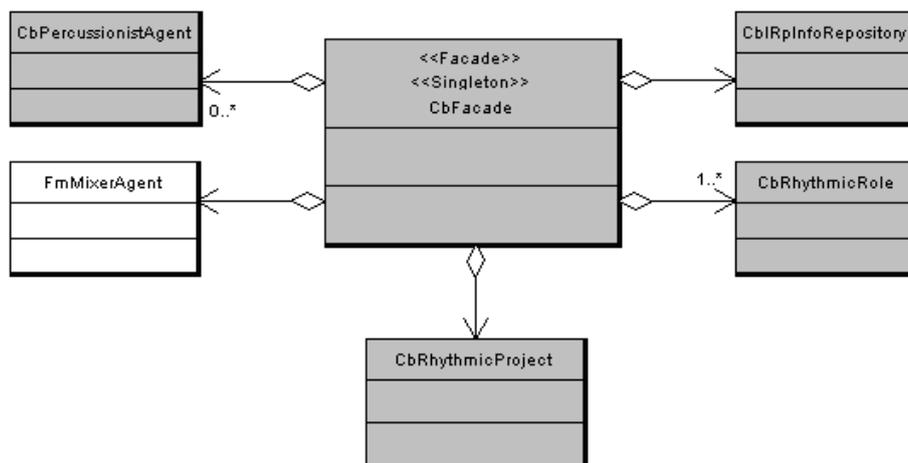


Figura 6.5: Fachada do sistema *CInBalada*

A classe *CbFacade* guarda a instância do repositório de padrões do sistema (implementação da interface *CbIRpInfoRepository*) e as instâncias dos papéis

rítmicos disponíveis (subclasses de `CbRhythmicRole`). Quando o usuário requisita a inserção de um novo instrumento no sistema, a fachada consulta os padrões do instrumento no repositório e o papel rítmico fornecido pelo usuário e usa essas informações para criar uma instância de `CbPercussionistAgent`. Por meio dessa instância a fachada também pode alterar as configurações de papel rítmico e repertório do agente mesmo depois de sua criação.

A fachada também tem uma instância do agente *mixer* (classe `FmMixerAgent`), que serve para que a fachada ofereça as operações básicas de configuração da performance implementadas nesse agente, como: mudança de andamento, mudança da métrica, etc. Com relação à mudança de andamento, é a fachada que trata a conversão entre os adjetivos de andamento (mostrados na seção 5.5) e o valor de andamento em BPM que é esperado pelo agente *mixer* (conforme apresentado na seção 4.2).

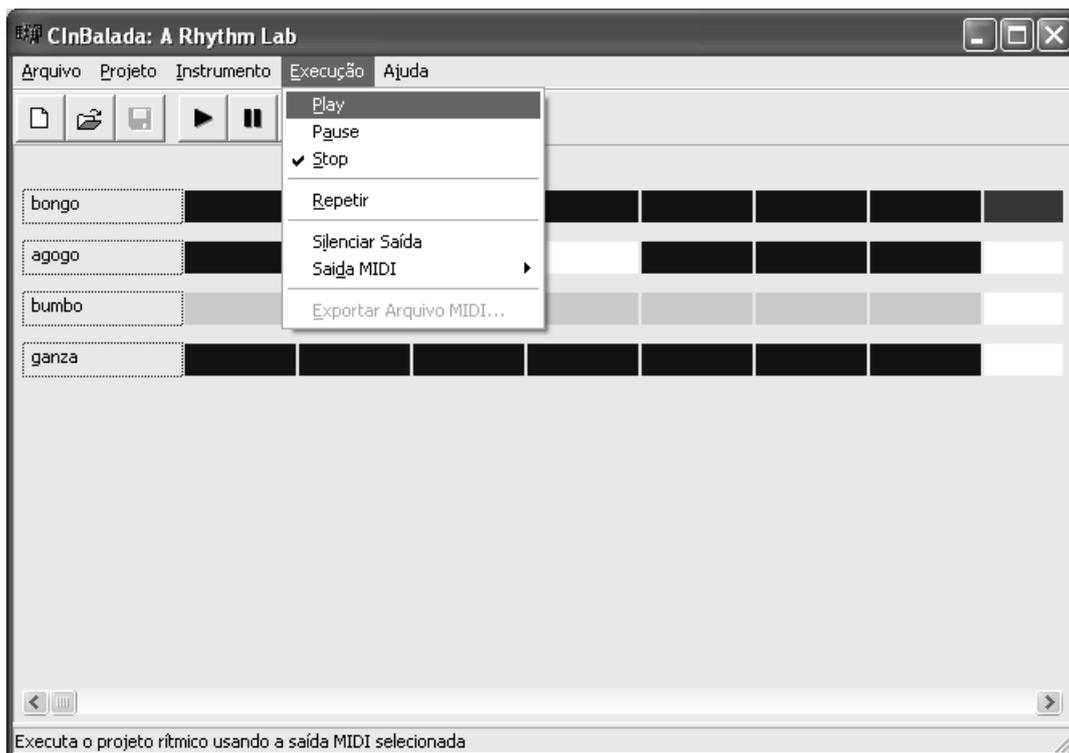
A classe `CbRhythmicProject` guarda algumas informações do estado da fachada, como: a métrica atual, os instrumentos (agentes) adicionados e os papéis rítmicos de cada instrumento. Essas informações podem ser salvas e carregadas de arquivo<sup>22</sup> pelo usuário do sistema.

## 6.5 Interfaces

O sistema `CInBalada` oferece duas interfaces gráficas para permitir o acesso às funcionalidades oferecidas pela fachada: a **interface principal** e o **editor de repertório**. Na interface principal (figura 6.6), o usuário tem acesso às funcionalidades de criação e execução de projetos rítmicos. Em outras palavras, o usuário pode setar parâmetros como: métrica, andamento, duração (em compassos), instrumentos e papel rítmico de cada instrumento a cada compasso (representados por quadrados coloridos na figura 6.6). Como resposta, o sistema gera uma composição rítmica, que pode ser controlada por comandos de *play*, *pause* e *stop*. Como comentamos na seção 5.3, a cada nova execução de um mesmo projeto (por exemplo, um comando *stop* seguido de *play*), o sistema pode produzir uma saída distinta, porém aderente às configurações definidas no projeto rítmico.

---

<sup>22</sup> A extensão adotada para arquivos de projeto rítmico foi “.cib”.



**Figura 6.6:** Interface principal do CInBalada

O editor de repertório do CInBalada oferece, basicamente, as funcionalidades de manipulação da base de padrões rítmicos do sistema. A interface agrupa as funcionalidades em diversas abas, como pode ser visto na figura 6.7. Na aba **Files**, o usuário pode adicionar e remover arquivos MIDI à base de padrões. Porém, o arquivo só será usado no sistema se as informações individuais do padrão (relatadas na seção 5.4) forem cadastradas na aba **Edit Pattern**. A aba **Instruments** permite cadastrar novos instrumentos com as informações mostradas na seção 5.4. Já a aba **MIDI Info** mostra informações de cada arquivo MIDI. Por fim, a aba **Comparator**, criada especialmente para auxiliar na depuração do sistema, permite ao usuário visualizar os padrões rítmicos em três representações distintas: TUBS, TEDAS e a representação poligonal. Além disso, esta aba também exibe, para cada padrão, os valores de todas as medidas de propriedades rítmicas implementadas, tais como: igualdade de espaçamento por regressão linear, atividade, *off-beatness*, imparidade rítmica e derivação.

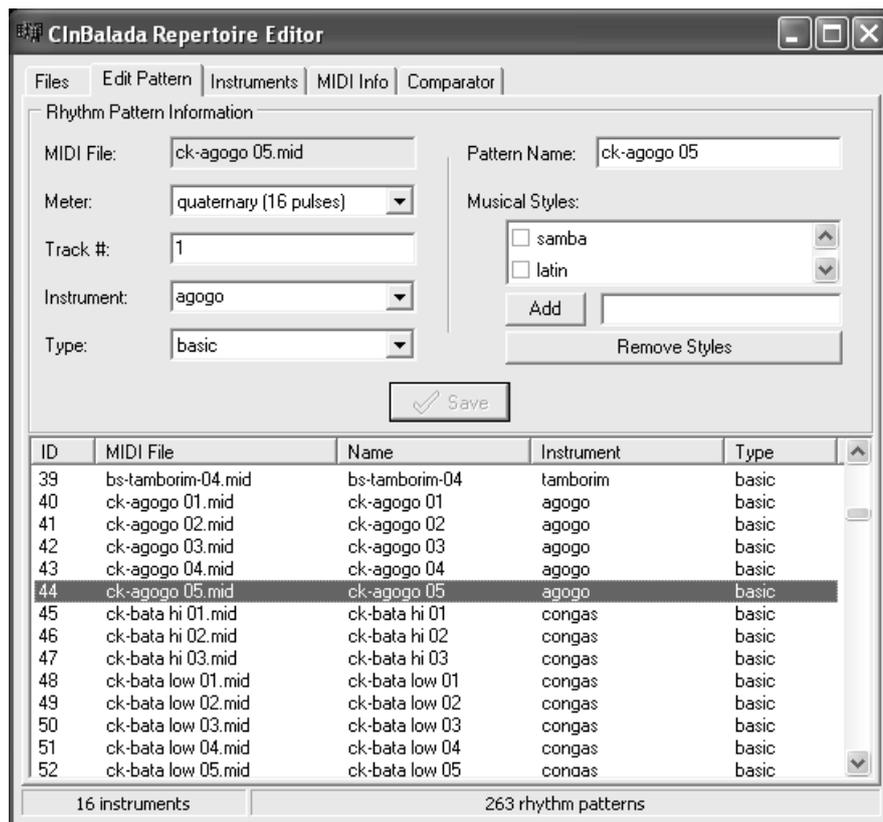


Figura 6.7: Editor de Repertório do CInBalada

## 6.5 Considerações Finais

O objetivo deste capítulo foi apresentar como o sistema CInBalada foi implementado. Para isso, mostramos os diagramas de classes de diferentes partes do sistema, explicando como essas classes realizam a modelagem explicada no capítulo 4. Uma vez que toda a construção do sistema já foi apresentada, tentaremos avaliar, no próximo capítulo, se os objetivos do sistema foram realmente alcançados.

## CAPÍTULO 7

# Avaliação do Sistema

Neste capítulo, avaliamos o sucesso do CInBalada no cumprimento do objetivo proposto de apresentar as características de *qualidade e diversidade*. Segundo afirmam Pearce (2000) e Papadopoulos & Wiggins (1999), são dois os principais erros de avaliação encontrados em alguns dos trabalhos musicais anteriores: a avaliação subjetiva pelo próprio autor e a escolha de avaliadores despreparados. Para avaliar o CInBalada quanto à satisfação dos seus requisitos de forma mais isenta e confiável, foram elaborados dois experimentos: um de avaliação da *qualidade*, discutido na seção 7.1, e outro de avaliação da *diversidade* apresentado na seção 7.2.

### 7.1 Avaliação de Qualidade

A noção de *qualidade* é certamente um requisito fundamental do sistema, porém ela é muito difícil de ser formalmente caracterizada em domínios artísticos de um modo geral. A qualidade, nessa classe de domínios, não está relacionada à conceitos como eficiência, precisão ou outras características formalmente exprimíveis, mas depende de valores estéticos inerentemente abstratos como *emoção*, *riqueza* (dos elementos estéticos presentes) e *harmonia* (equilíbrio entre os elementos)<sup>23</sup>.

Por esse motivo, a avaliação de qualidade no CInBalada não pôde ser medida diretamente. Para estimá-la, foi elaborado um experimento em que participantes humanos avaliaram a qualidade das saídas geradas pelo sistema segundo seus critérios estéticos pessoais. Durante a elaboração do experimento, a preocupação inicial que tivemos foi com a capacidade dos participantes de fazerem as avaliações de amostras individuais em termos absolutos (e.g. dizer que uma amostra é “boa”, simplesmente). O principal motivo para rejeitar o julgamento absoluto foi que a qualidade final da saída do sistema depende da base de padrões. Uma base pequena e formada apenas com padrões ruins não

---

<sup>23</sup> No domínio musical, especificamente, alguns dos termos usados são: *suinguado*, *alegre*, *triste* (e outros adjetivos de emoção) e até termos sinestésicos como *colorido*, *com brilho*, etc.

permitiria a geração de bons resultados, independentemente do mecanismo utilizado no sistema para escolha dos padrões. De fato, nem todos os padrões da base do CInBalada tem certas características de expressividade, como dito anteriormente (seção 5.4).

Por isso, decidimos fazer o experimento com base na comparação entre amostras geradas por três algoritmos de composição distintos, a partir de uma mesma base de padrões rítmicos. O algoritmo mais simples consiste na escolha dos padrões de cada instrumento de maneira aleatória (randômica), sem usar qualquer critério de avaliação. O segundo mecanismo usa a medida de distância cronotônica como único critério de avaliação na escolha dos padrões. O terceiro mecanismo avaliado é o mecanismo multi-critério baseado em papéis rítmicos usado no CInBalada, que consitui-se no principal objeto de avaliação do experimento.

As subseções a seguir descrevem o perfil dos participantes, a preparação do experimento e os resultados.

### **7.1.1 Participantes**

A literatura recomenda que experimentos de avaliação de sistemas musicais de composição sejam feitos preferencialmente com músicos experientes e cita dois motivos principais para isso. O primeiro é que os músicos podem ser mais capazes de identificar diferenças entre amostras musicais e o segundo motivo é que eles são mais propensos a preferir amostras mais atípicas ou inovadoras (Papadopoulos & Wiggins, 1999; Peace, Winterstein & Colton, 2001).

Por outro lado, como a música é um fenômeno universal que pode ser apreciado por qualquer pessoa, a opinião do público geral (i.e. não-músicos) não pode deixar de ser considerada. Por esse motivo, procuramos participantes que se encaixassem nesses dois perfis: músicos e não-músicos. Para identificar o perfil dos participantes, foi utilizado um questionário (Anexo A) com perguntas sobre a ligação do participante com a música e sobre o gosto dele por ouvir novidades. O questionário não tem campo para colocar o nome do participante, pois isso poderia constrangê-lo de algum modo e influenciar nas suas respostas (fazendo avaliações muito favoráveis, por exemplo).

O grupo dos músicos foi composto por cinco pessoas, com tempo médio de experiência com música de 17 anos. Todos eles demonstraram gosto por ouvir sons novos (as duas últimas alternativas da última pergunta). Já no grupo dos não-músicos, colocamos cinco pessoas que não tocam instrumento ou que alegaram prática informal e

amadora. Esse grupo demonstrou um gosto por novidades mais brando (dois dos cinco não demonstraram interesse).

### 7.1.2 Preparação do Experimento

Para realização do experimento, foram elaborados seis “projetos rítmicos” no sistema CInBalada. Cada projeto rítmico consistia de quatro ou oito compassos, com os instrumentos escolhidos dentre aqueles que tinham mais padrões disponíveis na base, pois instrumentos de poucos padrões reduziriam o espaço de busca e tornariam os resultados homogêneos. Na tabela a seguir mostramos a orquestração adotada em cada projeto.

<i>Rodada</i>	<i>Orquestração</i>
1	cowbell e maracás
2	bumbo, bongô, agogô e ganzá
3	bumbo, cowbell, agogô, apito e maracas
4	bumbo, cowbell, agogô e congas
5	claves, ganzá e bongô
6	surdo e bongô

**Tabela 7.1:** Orquestração em cada rodada do experimento de qualidade

Para cada projeto rítmico, o sistema CInBalada foi executado uma única vez e a saída resultante foi salva<sup>24</sup>. Usando a mesma orquestração, também foram salvas as saídas da primeira execução de cada um dos outros mecanismos considerados (o randômico e o baseado em similaridade). Assim, para cada orquestração, foram separadas três amostras MIDI, sendo uma de cada mecanismo considerado. Para tornar a audição mais independente do equipamento (i.e. placa de som) usado no experimento, as amostras foram sintetizadas em uma placa SoundBlaster Live! e convertidas para o formato *wave*, de forma que o experimento pudesse ser realizado em qualquer computador.

As amostras de cada orquestração foram separadas de modo a organizar o experimento em seis rodadas (de três audições). Para cada rodada foi criado um questionário (Anexo B) com espaço para avaliar individualmente as amostras com

---

<sup>24</sup> As amostras geradas pelo CInBalada para o experimento de qualidade estão disponíveis no site do projeto: <http://www.cin.ufpe.br/~pas/cinbalada>.

conceitos que iam de “péssimo” a “excelente”. O questionário também pede para identificar qual a melhor e qual a pior das amostras de cada rodada. Um espaço livre, de preenchimento opcional, também foi disponibilizado para que o participante pudesse fazer comentários sobre essa escolha da melhor e da pior amostra.

### 7.1.3 Realização do Experimento

No momento de realização do experimento, foi explicado aos participantes que o propósito do experimento seria o de avaliar a qualidade de três mecanismos de composição baseados em padrões prontos. Para evitar que houvesse qualquer desvio nos resultados, os participantes não foram informados no início do experimento sobre o funcionamento dos três mecanismos nem sobre qual amostra pertencia a cada mecanismo. A principal instrução dada foi para que escolhessem qual das amostras mais agradava a cada rodada.

Os experimentos foram conduzidos em sessões individuais, com exceção de uma sessão que contou com três participantes ouvindo simultaneamente e respondendo individualmente. Em todas as sessões, os participantes tinham liberdade para ouvir as amostras na ordem que quisessem e quantas vezes quisessem, até ter segurança para responder ao questionário. Também não foi imposto qualquer limite de tempo para conclusão do experimento.

### 7.1.4 Resultados

A tabela 7.2 mostra os resultados do experimento considerando tanto participantes músicos como não-músicos.

<b>Amostras Avaliadas como “Melhores”, Por Rodada</b>				
	Randômico	Similaridade	CInBalada	
Rodada 1	<b>0%</b>	<b>20%</b>	<b>80%</b>	100%
Rodada 2	<b>40%</b>	<b>30%</b>	<b>30%</b>	100%
Rodada 3	<b>0%</b>	<b>30%</b>	<b>70%</b>	100%
Rodada 4	<b>20%</b>	<b>0%</b>	<b>80%</b>	100%
Rodada 5	<b>20%</b>	<b>30%</b>	<b>50%</b>	100%
Rodada 6	<b>0%</b>	<b>50%</b>	<b>50%</b>	100%
<b>Média</b>	<b>13,3%</b>	<b>26,7%</b>	<b>60,0%</b>	100%

Tabela 7.2: Experimento de qualidade – geral

De modo geral, o mecanismo baseado em papéis mostrou clara vantagem sobre os outros, tendo suas amostras escolhidas 60% das vezes como melhores da rodada. Apenas

na rodada 2 ele não foi o mecanismo dominante, tendo sua amostra escolhida como melhor apenas por 30% dos participantes, contra 40% do mecanismo randômico e 30% do baseado em similaridade.

As estatísticas separadas por grupo de participantes podem ser conferidas nas tabelas 7.3 e 7.4. A análise dessas tabelas mostra que os resultados do CInBalada são melhores para o grupo dos músicos do que para o grupo dos não-músicos, com exceção apenas da segunda rodada que apresentou um péssimo resultado naquele grupo.

<b>Amostras Avaliadas como “Melhores”, Por Rodada</b>				
	Randômico	Similaridade	CInBalada	
Rodada 1	<b>0%</b>	<b>0%</b>	<b>100%</b>	100%
Rodada 2	<b>60%</b>	<b>40%</b>	<b>0%</b>	100%
Rodada 3	<b>0%</b>	<b>20%</b>	<b>80%</b>	100%
Rodada 4	<b>0%</b>	<b>0%</b>	<b>100%</b>	100%
Rodada 5	<b>40%</b>	<b>20%</b>	<b>40%</b>	100%
Rodada 6	<b>0%</b>	<b>40%</b>	<b>60%</b>	100%
<b>Média</b>	<b>16,7%</b>	<b>20,0%</b>	<b>63,3%</b>	100%

**Tabela 7.3:** Experimento de qualidade – grupo dos músicos

<b>Amostras Avaliadas como “Melhores”, Por Rodada</b>				
	Randômico	Similaridade	CInBalada	
Rodada 1	<b>0%</b>	<b>40%</b>	<b>60%</b>	100%
Rodada 2	<b>20%</b>	<b>20%</b>	<b>60%</b>	100%
Rodada 3	<b>0%</b>	<b>40%</b>	<b>60%</b>	100%
Rodada 4	<b>40%</b>	<b>0%</b>	<b>60%</b>	100%
Rodada 5	<b>0%</b>	<b>40%</b>	<b>60%</b>	100%
Rodada 6	<b>0%</b>	<b>60%</b>	<b>40%</b>	100%
<b>Média</b>	<b>10,0%</b>	<b>33,3%</b>	<b>56,7%</b>	100%

**Tabela 7.4:** Experimento de qualidade – grupo dos não-músicos

No campo para comentários livres nas fichas, os participantes fizeram os seguintes comentários sobre a amostra do CInBalada usada na rodada 2: “vazio em algumas partes” e “final estranho”. Na verdade, esses dois comentários estão diretamente relacionados às próprias características do projeto rítmico que gerou essa amostra, pois o projeto foi criado com o agogô alternando-se nos papéis *base complementar* e *mudo* e com o bongô fazendo o papel *virada* no final. Aparentemente, o problema foi que os músicos não gostaram da própria concepção do projeto. Se essa rodada fosse ignorada, a preferência geral dos músicos pelo mecanismo do CInBalada subiria para 76,0%, enquanto que para o grupo dos não-músicos ficaria quase inalterada com 56,0%.

Nas demais rodadas, as amostras do CInBalada receberam comentários mais elogiosos dos participantes:

- Rodada 1: “arranjo é mais rico” e “[tem] mais swing”.
- Rodada 3: “bom equilíbrio entre os instrumentos” e “ótimo balanço”.
- Rodada 4: “muito swing” e “grande diferença [em relação às outras amostras]”.
- Rodada 5: “o balanço é melhor, dá vontade de dançar”.
- Rodada 6: “o que possui mais swing”.

É interessante notar, em qualquer das três tabelas, como a quantidade de informação usada por cada mecanismo para criar as amostras está em proporção direta com as estatísticas de preferência alcançadas pelo mecanismo. O randômico, que não usa qualquer critério, apresenta as piores estatísticas com 13,3% da preferência geral (tabela 6.2), enquanto o mecanismo baseado apenas no critério de similaridade apresenta resultados melhores com 26,7%. Já o mecanismo do CInBalada, baseado em múltiplos critérios, tem mais do que o dobro da estatística de preferência do mecanismo baseado em similaridade, ficando com 60%. Isso mostra como a adição de estrutura na música permite alcançar melhores resultados artísticos, conforme observado na seção 3.1.

Desse modo, vemos que, comparativamente, o CInBalada foi claramente o mecanismo que gerou as amostras de melhor valor estético. Além disso, o fato das amostras do CInBalada provocarem certas respostas emocionais nos participantes são um forte indicativo de que as amostras do sistema têm boa qualidade absoluta, podendo ser aproveitadas diretamente em composições maiores. Portanto, o experimento demonstra que o objetivo de produzir amostras de boa qualidade musical no CInBalada foi atingido.

## **7.2 Avaliação da Diversidade**

O sistema CInBalada pode apresentar dois níveis distintos de diversidade. Como apresentamos no capítulo 5, o sistema permite gerar arranjos distintos a cada execução de um mesmo projeto rítmico (*diversidade intraprojeto*). Além disso, o sistema permite criar certa diversidade de projetos rítmicos e, conseqüentemente, uma diversidade de arranjos rítmicos entre esses projetos (*diversidade interprojeto*). A diversidade intraprojeto é claramente menor do que a diversidade interprojeto (é um subconjunto). Portanto, a maneira mais fácil de confirmar a capacidade do sistema de gerar uma grande diversidade de arranjos seria comprovando a diversidade intraprojeto (pois, automaticamente, já

implicaria numa diversidade interprojeto). Assim, o foco do experimento foi na diversidade gerada por um projeto rítmico específico.

Duas abordagens distintas foram levadas em consideração para avaliar a diversidade intraprojeto do CInBalada. A primeira consistiria em avaliar diretamente a diversidade do sistema usando alguma medida de similaridade entre as saídas do sistema. A vantagem dessa abordagem é que ela permitiria aplicar técnicas estatísticas mais facilmente para medir a diversidade do sistema (*diversidade estatística*). O problema dessa abordagem é que, apesar de existirem várias medidas de similaridade para padrões rítmicos monofônicos (como visto na seção 3.3), ainda não existem medidas semelhantes para padrões polifônicos.

Já a segunda abordagem consiste em avaliar a *diversidade percebida* por meio de experimentos com humanos. O uso de participantes humanos para avaliar diversidade pode, ao mesmo tempo, ser uma vantagem e uma desvantagem dessa abordagem. Vantagem porque seus resultados são mais relevantes do que os resultados de alguma análise estatística da diversidade, já que música é um fenômeno cognitivo humano. A desvantagem é que é difícil para humanos fazer avaliações de um número grande de amostras, pois o processo se torna cansativo e impreciso<sup>25</sup>. Diante das vantagens e desvantagens de ambas as abordagens, o ideal seria avaliar tanto a *diversidade estatística* como a *diversidade percebida* do sistema. Porém, diante da ausência das medidas de similaridade necessárias para a primeira alternativa, apenas a segunda foi avaliada, por meio de um experimento descrito nas próximas seções.

Analogamente à discussão feita sobre a avaliação de qualidade na introdução da seção 7.1, acreditamos que é difícil avaliar de maneira *absoluta* a diversidade. Um dos motivos é que a diversidade final do sistema depende da base de padrões, que deve ser grande o suficiente para dar muitas opções de escolha aos agentes. Assim, optamos por fazer um experimento que avaliasse comparativamente a diversidade do sistema com alguma outra referência. A referência escolhida foi o mecanismo randômico, já relatado na seção 7.1, pois ele é capaz de gerar o máximo de diversidade possível para uma dada base de padrões. Portanto, o experimento foi modelado para avaliar a diversidade percebida do CInBalada em comparação com a diversidade percebida do mecanismo de escolha randômico. O que o experimento busca mostrar é que a diversidade percebida do

---

<sup>25</sup> Semelhantemente ao *fitness bottleneck*, comentado na seção 2.1, dos sistemas que usam avaliação humana no próprio processo de composição.

CInBalada é intermediária entre a total ausência de diversidade (todos os arranjos são similares) e a diversidade percebida do mecanismo randômico. As subseções a seguir detalham o experimento.

### **7.2.1 Participantes**

Os participantes foram exatamente os mesmos do experimento de avaliação da qualidade (vide subseção 7.1.1).

### **7.2.2 Preparação do Experimento**

Como o objetivo do experimento é medir apenas a diversidade intraprojeto, foi escolhido um único projeto rítmico dentre aqueles elaborados para o experimento da seção 7.1. O projeto foi executado oito vezes consecutivas e as oito amostras foram salvas. Aproveitando a mesma orquestração do projeto, foram salvas também as oito primeiras amostras geradas pelo mecanismo randômico (explicado na seção 7.1). Como no primeiro experimento, todas as amostras foram convertidas para o formato *wave* de forma que o experimento pudesse ser realizado em qualquer sistema.

Os dois grupos de oito amostras foram salvos separadamente para serem apresentados em duas rodadas distintas do experimento. Um questionário foi criado (Anexo C) com um espaço para avaliar coletivamente a diversidade percebida no grupo de oito amostras com notas de 0 (amostras são idênticas) a 10 (amostras são completamente distintas), a cada rodada. Um espaço livre, de preenchimento opcional, também foi disponibilizado para que o participante pudesse fazer comentários sobre a nota dada. Ao final das duas rodadas, uma pergunta foi colocada no questionário para o participante escolher explicitamente qual o grupo de amostras que apresentou mais diversidade.

### **7.2.3 Realização do Experimento**

O experimento de diversidade foi conduzido, com cada participante, na mesma sessão do primeiro experimento. No momento de realização do experimento, foi explicado aos participantes que o propósito do experimento seria o de avaliar a diversidade interna de dois conjuntos de arranjos. A cada rodada, os usuários tinham a liberdade de ouvir as amostras na ordem que quisessem e quantas vezes quisessem até ter segurança para responder ao questionário. Para responder a última pergunta, sobre qual o conjunto mais diverso, os usuários também ficaram livres para ouvir novamente cada conjunto até fazer a escolha.

## 7.2.4 Resultados

Na tabela 7.5, mostramos os resultados do experimento considerando todos os participantes. Como já era esperado, o conjunto gerado pelo mecanismo randômico apresentou a maior diversidade percebida, tendo sido escolhido o mais diverso por 90% dos participantes em geral. Ao analisar a nota média de diversidade de cada conjunto, percebe-se que a diferença de diversidade não foi tão grande – proporcionalmente, a nota de diversidade recebida pelo CInBalada é 72% da nota do mecanismo randômico.

	<i>Random</i>	<i>CInBalada</i>
Mais Diverso	<b>90%</b>	<b>10%</b>
Diversidade Média	<b>7,1</b>	<b>5,1</b>
Diversidade Média Proporcional	<b>100%</b>	<b>72%</b>

**Tabela 7.5:** Experimento de diversidade – geral

Como pode ser visto na tabela 7.6, a nota média atribuída pelo grupo dos músicos demonstram que este grupo percebeu uma diferença de diversidade um pouco maior entre os dois conjuntos de amostra. Isso já era ser esperado, pois músicos geralmente conseguem perceber diferenças musicais sutis mais facilmente, como comentamos no experimento anterior.

	<i>Não-Músicos</i>		<i>Músicos</i>	
	<i>Random</i>	<i>CInBalada</i>	<i>Random</i>	<i>CInBalada</i>
Mais Diverso	<b>100%</b>	<b>0%</b>	<b>80%</b>	<b>20%</b>
Diversidade Média	<b>6,8</b>	<b>5,2</b>	<b>7,4</b>	<b>5,0</b>
Diversidade Média Proporcional	<b>100%</b>	<b>76,5%</b>	<b>100%</b>	<b>68%</b>

**Tabela 7.6:** Experimento de diversidade – resultado separado por grupo

Por outro lado, o único dos participantes que escolheu o CInBalada como o mais diverso foi justamente um músico. Isso provavelmente aconteceu porque, quando requisitados a escolher o conjunto mais diverso, os participantes hesitavam em escolher o conjunto do mecanismo randômico por acharem suas amostras menos agradáveis. Em outras palavras, os participantes tinham dificuldade em avaliar *diversidade* sem considerar *qualidade*, como demonstra o comentário: “diversidade de padrões não muito agradáveis”. Além disso, alguns participantes reportaram que foi difícil fazer a escolha. Uma explicação para isso é que o grande número de amostras cansava os participantes.

Diante dos resultados do experimento, concluímos que o CInBalada consegue gerar uma diversidade de arranjos rítmicos não muito distante da diversidade gerada pelo mecanismo randômico, assumido como teto máximo. Portanto, o CInBalada atingiu o objetivo de produzir uma grande diversidade de composições rítmicas polifônicas.

### **7.3 Considerações Finais**

Neste capítulo, avaliamos o sucesso do CInBalada quanto à satisfação dos requisitos de *qualidade* e *diversidade*. Para cada um dos requisitos, foi realizado um experimento de validação. Em ambos os experimentos, músicos e não-músicos humanos ouviram amostras do sistema e as avaliaram por meio de questionários.

No experimento de qualidade, mostramos como, comparado com outros dois mecanismos de composição, o mecanismo usado no CInBalada foi claramente o que gerou as melhores amostras. No experimento de diversidade, mostramos que os participantes do experimento perceberam uma diversidade razoavelmente alta em um conjunto de amostras do sistema, tomando como referência máxima a diversidade do conjunto de amostras geradas aleatoriamente.

Portanto, concluímos que os dois requisitos do sistema foram satisfeitos.

## CAPÍTULO 8

# Conclusão

Este trabalho apresentou o sistema de composição rítmica CInBalada. Para contextualizar esse sistema dentro da área de pesquisa de Inteligência Artificial aplicada à Computação Musical, analisamos, no capítulo 2, outros sistemas musicais de composição, assumindo sistemas de composição como sistemas que combinam “partes” musicais em “todos” maiores. Observamos, naquele capítulo, que as pesquisas da área têm privilegiado a geração de melodia e harmonia em detrimento dos sistemas compositores de saídas puramente rítmicas. Analisando os (poucos) sistemas rítmicos existentes, identificamos que eles deixavam muitas possibilidades de trabalhos originais.

No capítulo 3, apresentamos um amplo levantamento bibliográfico de conceitos e técnicas cognitivamente relevantes que pudessem ser aplicadas na construção de um sistema musical. O capítulo em questão apresentou os elementos estruturais rítmicos, alternativas de representação desses elementos e, por fim, diversas propriedades rítmicas que podem ser calculadas a partir das representações analisadas.

No capítulo 4, apresentamos o *framework* para sistemas musicais multiagentes que serviu de base para o sistema CInBalada. O *framework* é formado por dois módulos: um para desenvolvimento de sistemas multiagentes (de propósito geral) e outro para gerenciamento das saídas sonoras MIDI. Apesar de ter sido desenvolvido para o CInBalada, o *framework* é genérico suficiente para poder ser usado em outras aplicações musicais baseadas em agentes.

Nos capítulos 5 e 6, mostramos como o sistema CInBalada foi modelado e implementado usando uma abordagem alternativa às encontradas na literatura para o problema da composição rítmica polifônico. Assumimos, na construção do sistema, os requisitos de *qualidade* e *diversidade*. O sistema usa um conjunto de agentes percussionistas inteligentes e uma base de padrões monofônicos pré-gravados para gerar performances rítmicas polifônicas. A modelagem dos agentes foi baseada na noção de *papéis rítmicos*, proposta neste trabalho. Um papel rítmico define vários critérios para a escolha das linhas rítmica de cada instrumento de percussão e, portanto, foi usado na

elaboração de um mecanismo de avaliação descentralizado. Tal mecanismo é usado como modelo de decisão dos agentes em um protocolo de negociação que tem por objetivo a escolha do padrão rítmico de cada agente.

No capítulo 7, avaliamos o sucesso do CInBalada quanto à satisfação dos requisitos escolhidos. Para isso, foram elaborados dois experimentos: um para avaliar a *qualidade* das amostras geradas pelo sistema e outro para avaliar a *diversidade* de amostras produzidas. Nos experimentos, participantes humanos (divididos em músicos e não-músicos) ouviram amostras do sistema e as avaliaram por meio de questionários específicos. Os resultados do experimento mostraram que o sistema conseguiu atingir os dois objetivos propostos.

Diante de tudo que foi apresentado, discutimos na subseção 8.1, a seguir, as contribuições trazidas por este trabalho, enquanto, na subseção 8.2, discutimos as possibilidades de trabalhos futuros abertas por esta pesquisa.

### **8.1 Contribuições Principais e Lições Aprendidas**

Uma contribuição do presente trabalho é a noção de **papéis rítmicos**, proposta diante da falta de teorias que explicassem a estrutura global de uma performance rítmica polifônica. Apesar de alguns dos papéis propostos serem baseados em termos relativamente difundidos entre músicos (especialmente os papéis *base* e *solo*), não havia uma definição precisa nem abrangente para eles, pelo menos até onde temos conhecimento. Com base na noção de papéis rítmicos, também criamos uma maneira de caracterizar qualquer manifestação rítmica.

Além disso, a noção de papéis rítmicos serviu, no CInBalada, como um mecanismo abstrato de controle sobre as performances rítmicas. Usando os papéis rítmicos, usuários podem especificar a estrutura de alto nível da performance, sem perda da qualidade. Junto com outros parâmetros de configuração das performances (e.g. mudanças de orquestração, de andamento e de métrica), o CInBalada também permite gerar uma grande diversidade de saídas. Assim, acreditamos que o CInBalada pode servir como um **sistema para experimentação com arranjos rítmicos polifônicos**.

Acreditamos também que o presente trabalho valida o uso de **negociação multiagente para composição**. No CInBalada, o mecanismo de negociação tem a função de resolver dependências entre os agentes criadas pelos papéis rítmicos. Além disso, a flexibilidade oferecida pelo mecanismo de negociação foi o responsável pela diversidade do sistema (conforme apresentado no final do capítulo 4), enquanto o mecanismo de

avaliação baseada em papéis rítmicos (que funcionava como modelo de decisão dos agentes) garantiu a qualidade.

De modo geral, os bons resultados do CInBalada são uma provável consequência da preocupação que tivemos durante a pesquisa de representar e manipular estruturas rítmicas de relevância cognitiva. Essa preocupação pode ser observada especialmente no capítulo 3, em que apresentamos um amplo **levantamento bibliográfico**, ainda não compilado na literatura, de conceitos e técnicas cognitivamente relevantes que pudessem ser aplicadas na construção de um sistema de composição rítmica. Diante do sucesso do CInBalada, o presente trabalho reforça a idéia, apresentada naquele capítulo, de que a adição de estrutura musical permite criar sistemas musicais de boa qualidade.

Um outro possível motivo para o sucesso do sistema foi o fato de ele trabalhar com padrões rítmicos monofônicos pré-gravados, diferente de outros sistemas que tentam compor nota por nota. Isso facilita a criação de performances polifônicas interessantes, como pode ser percebido no experimento de avaliação de qualidade do sistema, no qual o mecanismo randômico (que combina padrões sem nenhum critério) conseguiu boas avaliações, em algumas rodadas. Assim, outra conclusão da presente pesquisa é que o **uso de partes prontas** reduz bastante a complexidade de geração de composições maiores.

Outras contribuições presentes neste trabalho são, também, as **novas medidas de avaliação** propostas. Em especial, a medida de derivação e a medida de interplay. Essas medidas foram propostas diante de certas necessidades surgidas durante o desenvolvimento do CInBalada que nenhuma das medidas propostas na literatura podia satisfazer.

Uma última provável contribuição do presente trabalho é o **framework para sistemas musicais multiagentes**, apresentado na seção 4 e construído tendo desempenho e padronização como principais objetivos. O sistema CInBalada serviu para garantir o sucesso do *framework* em atingir o seu primeiro objetivo, pelo menos.

## **8.2 Trabalhos Futuros**

O sistema CInBalada foi avaliado, no presente trabalho, quanto à satisfação dos requisitos de *qualidade* e *diversidade*. Estes dois requisitos foram propostos por considerarmos que são requisitos mínimos necessários para que o sistema funcione como uma ferramenta de experimentação rítmica a ser usada por músicos humanos. No entanto, um outro requisito desejável em ferramentas de experimentação não foi considerado na

avaliação do CInBalada: a **novidade** dos arranjos gerados. Como vimos no capítulo de molagem (cap. 5), o sistema permite que projetos rítmicos sejam criados alterando-se (e mesclando) os elementos de manifestações rítmicas (e.g. orquestração, papéis rítmicos e padrões rítmicos). Isso é um indício de que o sistema pode permitir que sejam rompidas as barreiras estilísticas de uma manifestação específica, criando-se arranjos *novos*. No entanto, a avaliação do CInBalada quanto ao requisito *novidade* requer, possivelmente, outro experimento que propomos como trabalho futuro.

Ainda com relação ao propósito de servir como um ambiente de experimentação, falta ao sistema um projeto de interface voltado ao usuário final. A interface deve oferecer, com o máximo de usabilidade, todas as funcionalidades previstas neste trabalho (e.g. seleção de orquestração e de papéis rítmicos, configurações básicas de performance e adição/remoção dos padrões rítmicos da base). Além disso, outras funcionalidades também podem ser desenvolvidas em extensões futuras do sistema. Uma funcionalidade nova que propomos é um **editor de papéis rítmicos**, por meio do qual usuários poderiam criar novos papéis rítmico selecionando as medidas que devem compor cada um dos quatro critérios.

Como apresentamos neste trabalho, o sistema CInBalada se baseia do reagrupamento de padrões rítmicos monofônicos para gerar saídas rítmicas polifônicas. Assim, a base de padrões do sistema tem um forte impacto na qualidade e na diversidade das saídas. Porém, na atual versão do sistema, a base apresenta algumas limitações que precisam ser tratadas em trabalhos futuros. Uma limitação da base atual é que alguns instrumentos possuem poucos padrões rítmicos disponíveis. Pesquisar mais padrões rítmicos seria a solução mais direta para o problema. Porém, alguns instrumentos presentes na base atual, mesmo contando com uma grande diversidade de padrões rítmicos, não possuem padrões que satisfaçam o *critério de elegibilidade* de certos papéis rítmicos pelo fato de o instrumento não ser usado para o papel em questão em nenhuma manifestação rítmica conhecida. Isso limita as possibilidades de uso do instrumento nos projetos rítmicos, pois, idealmente, qualquer instrumento deve ser capaz de assumir qualquer papel rítmico do sistema. Para maximizar o aproveitamento dos padrões cadastrados no sistema e permitir que qualquer instrumento assuma qualquer papel rítmico, uma idéia de extensão para o CInBalada seria implementar a **adaptação de padrões rítmicos** de um instrumento para outro, fazendo os padrões da base funcionarem como *templates* que podem ser aplicados a qualquer instrumento. Por exemplo, padrões de agogô cadastrados na base poderiam ser adaptados pelo sistema para serem usados

como padrão de surdo, permitindo que o surdo assumisse os papéis de base complementar e solo.

Como observado por Pearce & Wiggins (2001), um modelo computacional é, antes de tudo, um meio de comparar e comunicar teorias entre áreas de pesquisa diferentes. Nessa perspectiva, podemos dizer que o sistema CInBalada foi um tentativa de **validar a noção de papéis rítmicos**, comunicando Musicologia com a área de pesquisa que estuda composição automática. Diante do sucesso do sistema, podemos dizer que o conceito proposto foi parcialmente validado. Para uma validação completa do conceito de papéis rítmicos, seria necessário um estudo musicológico mais profundo sobre as performances rítmicas polifônicas. Semelhantemente, as novas medidas de avaliação propostas neste trabalho (a medida de derivação e a medida de *interplay*) precisam ter sua validade musical comprovada em trabalhos futuros.

Outra melhoria importante em versões futuras do sistema seria a preparação de uma base de padrões rítmicos com **expressividade**. Para isso, seria necessário seqüenciar os padrões rítmicos diretamente a partir de um instrumento de percussão MIDI tocado por um músico humano experiente. Uma abordagem alternativa seria continuar a utilizar padrões extraídos diretamente de partituras e adicionar expressividade a esses padrões usando alguma técnica de IA. Porém, trabalhos anteriores, como o de Bilmes (1993), mostram que esse é um problema bastante complexo.

Com relação aos papéis rítmicos do sistema, toda a modelagem proposta na seção 5.6 consiste apenas em uma das alternativa possíveis. A partir das mesmas definições apresentadas na seção 4.2, **outras possíveis modelagens** podem ser propostas futuramente, possivelmente com base em uma pesquisa musicológica mais profunda e usando outras medidas de avaliação.

Uma alternativa ao uso de medidas de propriedades rítmicas (conhecimento simbólico), seria implementar cada critério dos papéis rítmicos por meio de algum mecanismo de **aprendizagem de máquina**, como redes neurais (conhecimento sub-simbólico). As representações dos padrões rítmicos discutidas na seção 3.2 são algumas das idéias de entradas para redes neurais funcionando como critério de um papel rítmico. Porém, uma abordagem não encontrada na literatua que pode dar bons resultados seria usar as medidas de propriedades rítmicas como entradas para uma rede neural (uma abordagem mista de conhecimento simbólico e sub-simbólico).

Com relação ao *framework* proposto no capítulo 4, o problema de **padronização** ainda merece maior atenção. A criação de uma linguagem padrão para conteúdo das

mensagens de sistemas musicais ainda é um problema em aberto que merece ser analisado. Essa linguagem permitiria a interação entre agentes de diferentes projetistas, criando possibilidades mais interessantes.

Por fim, propomos como trabalho futuro uma extensão para permitir que o sistema **interaja com músicos humanos** para acompanhá-los em uma performance ao vivo. Essa ferramenta serviria para fazer o papel de um ou mais percussionistas tocando com um grupo de músicos humanos, podendo ser usada em uma apresentação ao vivo (no palco), em um ensaio em estúdio, ou até mesmo para um único músico praticar sozinho. A ferramenta poderia ser configurada (segundo parâmetros a serem definidos) por uma interface gráfica, antes do início da performance. Após o início da performance, as interfaces usadas pelos músicos para interagir com o sistema seriam apenas instrumentos MIDI ligados ao sistema.

# Bibliografia

- Arom, S. (1991). *African Polyphony and Polyrhythm*. Cambridge (England): Cambridge University Press.
- Bellifemine, F., Rimassa, G., & Poggi, A. (1999). JADE - A FIPA-compliant Agent Framework. Proceedings of the 4th International Conference and Exhibition on The Practical Application of Intelligent Agents and Multi-Agents, London (UK).
- Bertirotti, A. (2004). Temporal Dimension in Music. *PsicoLAB.net*, 0. Acesso 19 jan. 2006, na WWW: <<http://www.psicolab.net>>.
- Biles, J. A. (1994). GenJam: A Genetic Algorithm for Generating Jazz Solos. *Proceedings of the International Computer Music Conference 1994*, San Francisco (USA).
- Bilmes, J. (1992). A Model for Musical Rhythm. *Proceedings of the International Computer Music Conference 1992*, San Jose (USA), pp. 207-210.
- Bilmes, J. (1993). *Timing is of the Essence: Perceptual and Computational Techniques for Representing, Learning, and Reproducing Expressive Timing in Percussive Rhythm*. Dissertação (MSc), Massachusetts Institute of Technology, Cambridge, MA (USA).
- Block, S., Douthett, J. (1994). Vector products and intervallic weighting. *Journal of Music Theory*, 38, pp. 21-41.
- Burton, A. R. (1998). *A Hybrid Neuro-Genetic Pattern Evolution System Applied to Musical Composition*. Tese de PhD, University of Surrey (UK).
- Clough, J., Douthett, J. (1991). Maximally Even Sets. *Journal of Music Theory*, 35, pp. 93-173.

- Costalonga, L. L., Miletto, E. M., Flores, L. V., Alvares, L. O. C., Vicari, R. M. (2003). Um Sistema Multiagente para Simulação de Performance Rítmica no Violão. *Proceedings of the 9th Brazilian Symposium on Computer Music (SBCM 2003)*, Campinas, SP.
- Dannenberg, R. B. (1993). A Brief Survey of Music Representation Issues, Techniques, and Systems. *Computer Music Journal*, 17 (3), pp. 20–30.
- Díaz-Báñez, J. M., Farigu, G., Gómez, F., Rappaport, D., & Toussaint, G. (2004). El Compás Flamenco: A Phylogenetic Analysis. *Proceedings of BRIDGES: Mathematical Connections in Art, Music and Science*, Winfield (USA), pp. 61-70.
- Fonseka, J. R. (2000). *Musical Agents*. Honours Thesis (Monografia), Monash University (Australia).
- Fucks, W. (1962). Mathematical Analysis of Formal Structure of Music. *IRE Transactions on Information Theory*, 8, pp. 225-228.
- Ganascia, J. G., Ramalho, G., & Rolland, P. Y. (1999). An Artificially Intelligent Jazz Performer. *Journal of New Music Research*, 28 (2), pp. 105-129.
- Gómez, F., Melvin, A., Rappaport, D., & Toussaint, G. (2005). Mathematical Measures of Syncopation. *Proceedings of BRIDGES: Mathematical Connections in Art, Music, and Science*, Banff (Canada), pp.73-84.
- Goto, M., Hidaka, I., Matsumoto, H., Kuroda, Y., & Muraoka, Y. (1996). A Jazz Session System for Interplay Among All Players: VirJa Session (Virtual Jazz Session System). *Proceedings of the International Computer Music Conference 1996*, Hong Kong, pp. 346-349.
- Guéret, C., Monmarché, N., & Slimane, M. (2004). Ants can play music. *Proceedings of the Fourth International Workshop on Ant Colony Optimization and Swarm Intelligence*, Brussels (Belgium).
- Gustafson, K. (1988). The graphical representation of rhythm. *Progress Reports from Oxford Phonetics (PROPH)*, 3, pp. 6–26.

- Hofstadter, D. (1979). *Gödel, Escher, Bach: an Eternal Golden Braid*. Basic Books: Vintage Books.
- Honing, H. (1993). Issues in the representation of time and structure in music. *Contemporary Music Review*, 9, pp 221-239.
- Honing, H. (2002). Structure and Interpretation of Rhythm and Timing. *Tijdschrift voor Muziektheorie* [Jornal de Teoria Musical], 7 (3), pp. 227-232.
- Horowitz, D (1994). Generating Rhythms with Genetic Algorithms. *Proceedings of the 1994 International Computer Music Conference*, Aarhus (Denmark).
- Hsu, K. J., & Hsu, A. J. (1990). Fractal Geometry of Music. *Proceedings of the Natl. Acad. Sci. USA*, 87, pp. 938-941.
- Iyer, V., Bilmes, J., Wright, M., & Wessel, D. (1997). A Novel Representation for Rhythmic Structure. *Proceedings of the International Computer Music Conference '97*, Thessaloniki (Greece).
- Jennings, N. R., Faratin, P., Lomuscio, A. R. (2001). *Group Decision and Negotiation*, 10, pp. 199-215. (S.I): Kluwer Academic Publishers.
- Jewell, M. O., Middleton, L., Nixon, M. S., Prügel-Bennett, A., & Wong, S. C. (2005). A Distributed Approach to Musical Composition. *Proceedings of the 9th International Conference on Knowledge-Based & Intelligent Information & Engineering Systems (KES'2005)*, Melbourne (Australia).
- Keith, M. (1991). *From Polychords to Pólya: Adventures in Musical Combinatorics*. Princeton : Vinculum Press.
- Kelemen, J. (2003). The Agent Paradigm. *Computing and Informatics*, 22 (6), pp. 513-519.
- Kon, F. and Iazzetta, F. (1998). Internet music: Dream or (Virtual) Reality. *Proceedings of the 5th Brazilian Symposium on Computer Music (SBCM'98)*, Belo Horizonte, MG.
- Ladzekpo, C. K. (n.d.). *Rhythmic Principles*. Acesso 19 jan. 2006, na WWW: <<http://cnmat.cnmat.berkeley.edu/%7Eladzekpo/PrinciplesFr.html>>.

- Langen, P. H. G. van, Wijngaards, N. J. E., & Brazier, F. M. T. (2004). Designing Creative Artificial Systems. In: Duffy, A. H. B. & Brazier, F. M. T. (eds.), *AIEDAM Journal, Special Issue on Learning and Creativity in Design*, 18 (4), pp. 217-225.
- Lartillot, O. (2003). Discovering Musical Patterns through Perceptive Heuristics. *Proceedings of the 4th International Symposium on Music Information Retrieval (ISMIR 2003)*, Baltimore (USA), pp. 89–96.
- Lerdahl, F., & Jackendoff, R. (1983). *A Generative Theory of Tonal Music*. Cambridge, MA: MIT Press.
- Lima, R. (2005). *DANCE: Uma Infra-Estrutura Computacional para Agentes Dançarinos*. Dissertação de Mestrado, Centro de Informática, Universidade Federal de Pernambuco, Recife – PE.
- London, J. (2000). Hierarchical Representations of Complex Meters. *Proceedings of the 6th International Conference on Music, Perception and Cognition*, Keele University (UK).
- Mantaras, R. L. de, & Arcos, J. L. (2002). AI and Music: from Composition to Expressive Performance. *AI Magazine*, 23 (3), pp. 43-57.
- Maranhão, S. M., Jansen, S. B., Galvão, W. de C., Gomes, A. S., Araújo, E. P., & Cysneiros, P. G. (2003). Ensino de Maracatu de Baque Virado Através de Software Educativo. *Proceedings of 8º Taller Internaional de Software Educativo*, Santiago (Chile).
- McAdams, S. (1989). Psychological constraints on form-bearing dimensions in music. *Contemporary Music Review*, 4, pp. 181-198.
- McLachlan, N. (2000). A Spatial Theory of Rhythmic Resolution. *Leonardo Music Journal*, 10, pp. 61-67.
- Messick, P. (1998). *Maximum MIDI: Music Applications in C++*. Greenwich, CT: Manning Publications.
- Miranda, E. R. (2002). Emergent Sound Repertoires in Virtual Societies. *Computer Music Journal*, 26(2), pp. 77-90.

- Mitchell, M. (1996). *An Introduction to Genetic Algorithms*. Cambridge, MA: The MIT Press.
- Murray-Rust, D. (2003). *VirtuaLatin - Agent Based Percussive Accompaniment*. Dissertação (MSc in Informatics), School of Informatics, University of Edinburgh, Edinburgh (Scotland).
- Murray-Rust, D., & Smaill, A. (2005). Musical Acts and Musical Agents. *5th MUSICNETWORK Open Workshop*, Vienna (Austria).
- Pachet, F. (2000). Rhythms as Emerging Structures. *Proceedings of the International Computer Music Conference 2000*, Berlin (Germany).
- Pachet, F. (2002). Interacting with a musical learning system: the Continuator. In: C. Anagnostopoulou, M. Ferrand, and A. Smaill (eds.), *Lecture Notes in Artificial Intelligence*, 2445, pp. 119–132. (S.l.): Springer-Verlag.
- Papadopoulos, G., & Wiggins, G. (1999). AI Methods for Algorithmic Composition: A Survey, a Critical View and Future Prospects. *Proceedings of the AISB'99 Symposium on Musical Creativity*, Edinburgh (Scotland), pp. 110-117. Brighton, UK: SSAISB.
- Parncutt, R. (1994). A perceptual model of pulse salience and metrical accent in musical rhythms. *Music Perception*, 11, pp. 409-464. (S.l.): University of California Press.
- Pazos, A., Riego, A. S. del, Dorado, J., & Cardalda, J. R. (1999). Genetic Music Composer. *Proceedings of the Congress of Evolutionary Computation 1999 (CEC'99)*, Washington (USA), pp. 885-890.
- Pearce, M. (2000). *Generating Rhythmic Patterns: a Combined Neural and Evolutionary Approach*. Dissertação (MSc in Artificial Intelligence), School of Informatics, University of Edinburgh, Edinburgh (Scotland).
- Pearce, M. T. & Wiggins, G. A. (2001). Towards a framework for the evaluation of machine compositions. In *Proceedings of the AISB'01 Symposium on Artificial Intelligence and Creativity in the Arts and Sciences*, York (England), pp. 22–32. Brighton, UK: SSAISB.

- Pease, A., Winterstein, D., & Colton, S. (2001). Evaluating machine creativity. *Proceedings of the International Conference on Case Based Reasoning 2001*, Vancouver (Canada).
- Pfleiderer, M. (2003). The Study of Rhythm in Popular Music – Approaches and Empirical Results. *Proceedings of the 5th Triennial ESCOM Conference*, Hanover (Germany).
- Poslad, S., Buckle, P., & Hadingham, R. G. (2000). The FIPA-OS agent platform: Open Source for Open Standards. *Proceedings of PAAM 2000*, Manchester (UK), pp. 355-368.
- Pressing, J. (1997a). Cognitive Complexity and the Structure of Musical Patterns. *Proceedings of the Fourth Conference of the Australasian Cognitive Science Society*, Newcastle (Australia).
- Pressing, J. (1997b). Some Perspectives on Performed Sound and Music in Virtual Environments. *Presence*, 6 (4), pp. 1–22.
- Pressing, J. (2002). Black Atlantic Rhythm: Its Computational and Transcultural Foundations. *Music Perception*, 19 (3), pp. 285-310.
- Russel, S., & Norvig, P. (2004). *Inteligência Artificial* [Tradução de: Artificial Intelligence – A Modern Approach, 2<sup>nd</sup> ed]. Rio de Janeiro: Elsevier Editora.
- Salazar, M. (1991). *Batucadas de Samba*. Rio de Janeiro : Lumiar Editora.
- Sampaio, P. (2003). *Framework Multiagente Para Aplicações Musicais*. Trabalho de Graduação (Monografia), Centro de Informática, Universidade Federal de Pernambuco.
- Sandroni, C. (1996). Mudanças de padrão rítmico no samba carioca, 1917-1937. *Revista Transcultural de Música (Transcultural Music Review)*, 2.
- Santos, C. de O., Resende, T. S. (2005). *Batuque book maracatu: baque virado e baque solto*. Recife: (S.n).
- Searle, J. L. (1969). *Speech Acts*. Cambridge (UK): Cambridge University Press.

- Shmulevich, I., & Povel, D.J.L. (2000). Measures of Temporal Pattern Complexity. *Journal of New Music Research*, 29(1), pp. 61-70.
- Singh, P. (2003). Examining the Society of Mind. *Computing and Informatics*, 22 (6), pp. 521-523.
- Stroup, B. (1997). *The C++ Programming Language (Third Edition)*. Boston, MA : Addison-Wesley.
- Todd, P. M., & Werner, G. M. (1999). Franksteinian Methods for Evolutionary Music Composition. In: N. Griffith and P. M. Todd (eds.), *Musical Networks – Parallel Distributed Perception and Performance*, p. 313. Cambridge, MA: MIT Press.
- Todd, P. M., & Miranda, E. R. (2004). Putting some (artificial) life into models of musical creativity. In: I. Deliege and G. Wiggins (eds.), *Musical creativity: Current research in theory and practice*. (S.1): Psychology Press.
- Toussaint, G. (2002). A Mathematical Analysis of African, Brazilian and Cuban Clave Rhythms. *Proceedings of BRIDGES: Mathematical Connections in Art, Music and Science*, Towson (USA).
- Toussaint, G. (2003a). Algorithmic, Geometric, and Combinatorial Problems in Computational Music Theory. *Proceedings of X Encuentros de Geometria Computacional*, Sevilla (Spain), pp. 101-107.
- Toussaint, G. (2003b). Classification and Phylogenetic Analysis of African Ternary Rhythm Timelines. *Proceedings of BRIDGES: Mathematical Connections in Art, Music and Science*, Granada (Spain), pp. 25-36.
- Toussaint, G. (2004a). *A Comparison of Rhythmic Similarity Measures*. Technical report SOCS-TR-2004.6, School of Computer Science, McGill University (Canada).
- Toussaint, G. (2004b). The Geometry of Musical Rhythm. *Proceedings of the Japan Conference on Discrete and Computational Geometry (JCDCG 2004)*, pp. 198-212.
- Toussaint, G. (2005). Mathematical Features for Recognizing Preference in Sub-Saharan African Traditional Rhythm Timelines. *Proceedings of the 3rd International Conference on Advances in Pattern Recognition (ICAPR 2005)*, Bath (UK), pp. 18-27.

- Ueda, L. K. & Kon, F. (2004). Andante: Composition and performance with mobile musical agents. *Proceedings of the International Computer Music Conference 2004*, Miami (USA), pp. 604–611.
- Walker, W. F. (1997). A Computer Participant in Musical Improvisation. *Proceedings of the Conference on Human Factors in Computing Systems (CHI 97)*, Atlanta (USA).
- Weyde, T. (2001). Grouping, Similarity and the Recognition of Rhythmic Structure. *Proceedings of the International Computer Music Conference 2001*, Havana (Cuba).
- Wooldridge, M. J. (2002). *An Introduction to Multi-Agent Systems*. (S.I.): John Wiley and Sons.
- Wulfhorst, R. D., Nakayama, L. & Vicari, R. M. (2003). A Multiagent Approach for Musical Interactive Systems. *Proceedings of the Second International Joint Conference on Autonomous Agents and Multiagent Systems*, Melbourne (Australia).

## APÊNDICE A

# Questionário Sobre o Participante

- Como você definiria sua ligação com a música? (Pode marcar mais de um)
  - Ouvinte casual
  - Ouvinte entusiasta (audiófilo)
  - Músico amador
  - Músico profissional
  - Outro: .....
  
- Se músico, há quanto tempo? .....
- Qual(is) instrumento(s) toca? .....
  
- Quais os gêneros musicais que você mais gosta?  
.....
  
- Qual das alternativas abaixo descreve mais fielmente seu gosto por novidades na música (ou seja, seu gosto por ouvir sons que não conhecia)?
  - Ouço sempre as mesmas coisas (mesmos artistas)
  - Quando ouço coisas novas são sempre são dentro dos meus gêneros favoritos
  - Gosto de ouvir coisas novas dos meus gêneros favoritos e às vezes também me interesse por conhecer outros gêneros
  - Estou sempre buscando conhecer gêneros novos
  
- Comente (opcional): .....
- .....

## APÊNDICE B

# Questionário do Experimento de Qualidade

*Rodada [X]: Instrumento A, Instrumento B, ...*

- Avalie o arranjo rítmico **A**:  
 Péssimo    Ruim    Bom    Excelente
- Avalie o arranjo rítmico **B**:  
 Péssimo    Ruim    Bom    Excelente
- Avalie o arranjo rítmico **C**:  
 Péssimo    Ruim    Bom    Excelente
- Qual dos arranjos você mais gostou?  
 **A**    **B**    **C**
- Qual dos arranjos você menos gostou?  
 **A**    **B**    **C**
- Comente a escolha do melhor e do pior (opcional):  
.....  
.....  
.....

## APÊNDICE C

# Questionários do Experimento de Diversidade

### 1. Conjunto [X]: Instrumento A, Instrumento B, ...

- Que nota você daria, de 0 a 10, para avaliar a semelhança geral entre os arranjos rítmicos deste conjunto? (Onde 10 significa que os padrões são “totalmente distintos” e 0 significa que são “idênticos”).

0    1    2    3    4    5    6    7    8    9    10

(idênticos)

(totalmente  
distintos)

Comente (opcional): .....  
.....  
.....

---

### 2. Final

- Em geral, qual dos dois conjuntos apresenta maior diversidade?  
( ) Primeiro                      ( ) Segundo

Comente (opcional): .....  
.....  
.....