

# CInBalada: um Laboratório Multiagente de Geração de Ritmos de Percussão

Pablo Sampaio, Patrícia Tedesco, Geber Ramalho

Centro de Informática (CIn) – Universidade Federal de Pernambuco (UFPE)  
Caixa Postal 7851 – 50.732-970 – Recife – PE – Brasil

{pas,pcart,glr}@cin.ufpe.br

***Abstract.** In this paper we present CInBalada, a multiagent system where intelligent percussionist agents participate in purely rhythmic performances. Each agent knows a limited set of rhythmic patterns for a percussion instrument. Thus, agents interact with their peers to decide which patterns they should play so that an interesting group performance emerges. Human users can try different ensembles and musical mixes with CInBalada, turning the system into a Rhythms Lab, which may be useful for composers and other musicians.*

***Resumo.** Neste trabalho apresentamos o sistema CInBalada, onde agentes percussionistas inteligentes participam de uma performance puramente rítmica. Cada agente conhece um repertório limitado de padrões rítmicos de um instrumento de percussão e precisa interagir com os demais agentes para escolher qual padrão tocar com o grupo de maneira que a performance fique musicalmente boa. Usuários humanos podem usar o sistema para testar diferentes formações e diferentes misturas de estilos, fazendo o sistema funcionar como um laboratório rítmico, que pode vir a ser útil para músicos compositores e arranjadores.*

## 1. Introdução

Na intersecção entre as áreas de Inteligência Artificial e Computação Musical, vários sistemas foram criados aplicando o conceito de agentes inteligentes (Pachet, 2000) (Pachet, 2002) (Miranda, 2002) (Murray-Rust, 2003) (Wulfhorst, et.al 2003). Nesses sistemas, diversos aspectos da música são abordados, porém a ênfase maior costuma ser na geração de harmonia e/ou melodia. Poucos sistemas focam especificamente a parte rítmica, que é, entretanto, considerada fundamental para a “sensação” que a música causa em ouvintes humanos (Honning, 2002), além de ter grande importância cultural em algumas sociedades como algumas tribos da África (Ladzekpo, n.d).

Dentre os poucos sistemas inteligentes que abordam os instrumentos rítmicos, aparentemente a maior parte é voltada especificamente para bateria (Gannon, 2005) (Kragtwijk, et.al 2001) (Pearce, 2000), sendo ainda menor o número daqueles voltados para instrumentos de percussão em geral. Os poucos sistemas voltados para percussão não tratam de maneira homogênea da interação entre agentes inteligentes diversos e, quando o fizeram, consideraram um modelo de interação distante do que acontece com músicos reais. Além disso, não há sistemas inteligentes para a criação de arranjos puramente rítmicos, com possíveis misturas de estilos musicais.

Neste trabalho, apresentamos um sistema inovador voltado para percussão, o CInBalada, em que agentes percussionistas interagem para gerar uma performance puramente rítmica. Usuários do sistema poderão controlar as performances de cada um desses agentes de diversas maneiras: escolhendo os instrumentos presentes na performance, definindo um estilo musical (por agente) e alterando o andamento geral. Dessa forma, o CInBalada funciona como uma espécie de laboratório de ritmos, por meio do qual usuários (músicos, principalmente) poderão testar diferentes misturas rítmicas e criar arranjos de percussão variados. Além disso, tentamos abordar a questão da interação entre os agentes de uma maneira mais próxima ao que acontece em grupos musicais humanos de modo a permitir, futuramente, fazer do CInBalada um sistema interativo para uso profissional em shows ou como uma ferramenta de aprendizado para músicos iniciantes praticarem sozinhos.

Na próxima seção apresentamos os principais sistemas multiagentes existentes que tratam de performances rítmicas, bem como uma visão crítica desses sistemas. Na seção 3 apresentamos uma visão geral do sistema CInBalada. Na seção 4 apresentamos o agente responsável pela execução sonora do CInBalada: o agente *outputter*, e na seção 5 apresentamos detalhes dos agentes percussionistas do sistema. A seção 6 descreve a arquitetura e implementação do sistema. Por fim, a seção 7 apresenta conclusões e uma discussão sobre futuras extensões do CInBalada.

## **2. Percussionistas Virtuais**

Nesta seção analisaremos dois sistemas multiagentes que tratam da criação de performances rítmicas: o Rhythm Band Editor e o VirtualLatin.

### **2.1 Rhythm Band Editor (Pachet, 2000)**

No Rhythm Band Editor, cada agente começa com um padrão rítmico inicial (possivelmente vazio) e uma base de regras de transformação. Uma partitura global é mantida de modo a permitir que cada agente tenha acesso ao que os demais agentes estão tocando. Um módulo de percepção presente nos agentes recebe como entrada a partitura global completa 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 corrente do agente. As regras têm suas pré-condições especificadas a partir das informações de alto-nível geradas pelo módulo de percepção e, quando disparadas (pré-condições satisfeitas), executam uma ação de edição no ritmo 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”.

Os agentes do Rhythm Band Editor permanecem indefinidamente em ciclo, alternando, a execução de seus respectivos padrões com a aplicação das regras, de modo a evoluir os padrões rítmicos. Eventualmente, o sistema pode convergir para uma situação em que nenhuma regra é aplicável. Nesse caso, os agentes continuam a tocar seus padrões sem novas alterações.

### **2.2. VirtualLatin (Murray-Rust, 2003)**

Outro sistema voltado para percussão é o VirtualLatin, onde um agente inteligente, o timbaleiro, gera arranjos de timbales para salsa. O agente timbaleiro está

inserido num ambiente multiagente em que cada agente é um instrumentista de uma banda de salsa e produz saída musical de um compasso por vez. O timbaleiro, no entanto, é o único agente inteligente do sistema – os demais agentes apenas reproduzem trechos pré-gravados de uma música (do estilo salsa). Baseado no GTTM (Lerdahl & Jackendoff, 1983), o algoritmo do agente timbaleiro é capaz de extrair várias informações de alto nível da peça em execução 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 o padrão rítmico de salsa que melhor se adequa ao compasso, de acordo com o contexto musical em que ocorre. Por exemplo: o timbaleiro tende a executar viradas no final de cada seção da música.

### **2.3. Análise Crítica**

Os dois trabalhos apresentados na seção anterior tratam da construção de agentes percussionistas, porém apresentam algumas limitações, que apresentamos nesta seção.

O Rhythm Band Editor, apesar de permitir interações entre múltiplos agentes percussionistas apresenta a desvantagem de usar um modelo de interações um pouco distante do que acontece em performance reais. O sistema foi criado com o propósito principal de estudar os ritmos enquanto estruturas evolutivas e não o de criar saídas musicais que possam ser aproveitadas por músicos humanos. Portanto, o foco principal desse sistema acaba sendo o conjunto de regras e não a música produzida. Com isso, é muito difícil controlar o rumo da performance musical nesse sistema, pois não é possível ter uma clara idéia da música que vai emergir como resultado das regras escolhidas.

Já o VirtuaLatin implementa um único agente de percussão e produz arranjos típicos apenas de um estilo musical específico – a salsa. O sistema oferece, portanto, opções limitadas para criação de arranjos de percussão, servindo apenas para gerar arranjos de timbales partindo de salsas pré-gravadas (sem os timbales). Além disso, o sistema não apresenta flexibilidade nem homogeneidade no modelo de interação entre os agentes, pois existe apenas um agente inteligente que simplesmente recebe as saídas musicais dos demais agentes e as usa como entrada para gerar sua própria saída (o arranjo de timbales).

### **3. O CInBalada**

Neste contexto, o CInBalada oferece um ambiente inovador que oferece a possibilidade de experimentar as mais variadas misturas rítmicas, mesclando livremente estilos musicais usando as mais diversas formações de instrumentos de percussão. Desse modo, o sistema funciona como um laboratório rítmico, onde compositores e arranjadores podem criar e testar novos arranjos percussivos facilmente, sem ter que arcar com os custos de reunir músicos humanos para tocar em conjunto.

Inspirado em manifestações rítmicas populares (como o samba e o maracatu), o CInBalada dispõe de agentes inteligentes que simulam músicos percussionistas, que comunicam-se entre si para gerar uma performance rítmica polifônica. O uso de múltiplos agentes inteligentes é uma analogia natural para um grupo musical real tocando junto e permite a adaptação do sistema a cada nova situação imposta pelo

usuário (mudança de formação, mudança de estilo musical, etc.), além de permitir interatividade com músicos humanos numa possível extensão futura.

Para iniciar uma performance, o usuário **escolhe os instrumentos de percussão** que desejar. Para cada instrumento é associado um agente, possuidor de um conjunto (limitado) de padrões rítmicos do instrumento escolhido. Quando a performance é iniciada, os agentes tentarão escolher padrões rítmicos que se encaixem entre si (segundo uma função de avaliação), de modo a gerar uma performance musicalmente aceitável.

Esse processo de escolha dos padrões é realizado automaticamente pelos agentes, porém o usuário também tem a oportunidade de interferir nas escolhas ou de obrigar um agente a usar um padrão rítmico específico. Na atual versão, isso é feito por meio da **seleção de estilos musicais** de cada agente. O usuário pode selecionar um estilo musical específico para cada agente, de maneira a obrigar o agente a escolher apenas padrões rítmicos típicos dos estilos selecionados.

A figura 1 abaixo mostra a interface principal do sistema, onde podem ser acessadas as funcionalidades acima descritas. Nessa interface os usuários podem inserir e remover os agentes. Para inserir um agente, o usuário escolhe o instrumento do agente e um nome, que será usado para identificar o agente na interface. Atualmente, só pode haver um agente por instrumento de percussão. Na interface é possível visualizar uma tela de detalhes sobre cada agente. Na tela de detalhes é possível ver uma lista de estilos musicais que podem ser escolhidos, bem como o repertório completo de padrões rítmicos disponíveis para o dado agente.

A interface oferece, ainda, algumas opções básicas de **controle sobre a saída sonora** gerada. É possível parar (silenciar) e reiniciar a saída sonora, bem como alterar o andamento geral da performance a qualquer momento.

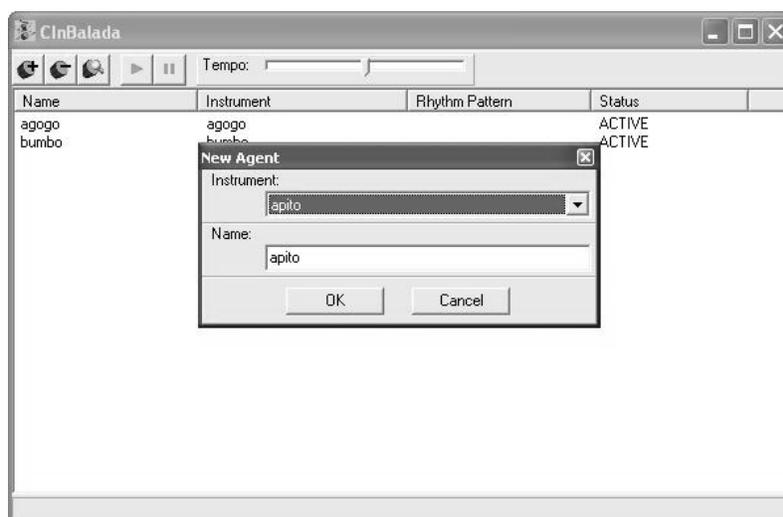


Figura 1. Interface principal do CInBalada

### 3.1. Repertório de Padrões Rítmicos

Os padrões rítmicos usados pelos agentes do sistema são armazenados como arquivos MIDI e possuem de um a quatro compassos. No momento, o sistema conta com uma

base de cerca de duzentos padrões rítmicos MIDI para doze instrumentos de percussão.

Cada um desses padrões rítmicos está associado a um conjunto de informações de mais alto nível, que são utilizadas pelos agentes do CInBalada em suas negociações.

As principais informações de alto nível armazenadas para cada padrão são:

- **Instrumento** usado no padrão (cada arquivo do sistema é executado por apenas um instrumento de percussão).
- **Estilos musicais** aos quais o padrão é comumente associado, ou seja, estilos musicais onde o padrão rítmico costuma ser usado.
- **Tipo** do padrão rítmico. Indica o papel que ele tem em uma performance. Atualmente, inspirados no sistema VirtualLatin, consideramos que os padrões podem ser de três tipos: *basic* (ritmo básico, deve ser tocado a maior parte do tempo), *fill* (virada) ou *roll* (rufo).
- **Assinatura de tempo**. Indica se o compasso é binário, ternário ou quaternário.
- **Arquivo MIDI**, que contém o padrão rítmico propriamente dito.

Os padrões rítmicos, juntamente com as informações de alto nível associadas, são cadastrados no sistema por meio de uma interface auxiliar chamada CInBalada Repertoire Editor, que pode ser vista na figura 2.



Figura 2. CInBalada Repertoire Editor.

#### 4. Agente *Outputter*

O agente *outputter* é responsável por receber e mesclar todas os trechos musicais (padrões rítmicos) enviados pelos agentes músicos e enviá-los a alguma saída (placa de

som ou arquivo). Existe apenas um agente *outputter* por instância da aplicação. Ele é criado automaticamente quando a aplicação é iniciada.

Quando um agente percussionista é criado, ele se cadastra junto ao agente *outputter* como **produtor de música**. A partir disso, o *outputter* passa a mandar para o agente músico, periodicamente, mensagens requisitando frases musicais novas. As frases podem ser formadas por um número qualquer de compassos completos e são transmitidas em formato XML. Um exemplo de uma frase musical no modelo XML adotado pode ser visto na figura 3.

```
<?xml version="1.0" encoding="utf-8"?>
<Phrase resolution="400">
  <Note pitch="36" dynamic="80" startTick="12" endTick="18"/>
  <Note pitch="43" dynamic="80" startTick="30" endTick="45"/>
</Phrase>
```

Figura 3. Exemplo do modelo XML adotado para representar frases musicais.

Para permitir a interação entre os agentes, é necessário também que um agente saiba o que será tocado pelos demais agentes. Para isso, o agente *outputter* também encaminha frases de um agente para outros. Um agente interessado em receber as frases de outro agente precisa enviar uma mensagem para se cadastrar como **ouvinte**, indicando na mensagem qual agente deseja ouvir. O *outputter* enviará cada nova frase do agente ouvido para os seus respectivos ouvintes.

O uso de um agente que centraliza a saída musical tem várias vantagens (Murray-Rust, 2003):

- Ponto central de comunicação. Implica na troca de  $\theta(n)$  mensagens, ao invés de  $\theta(n^2)$ , se cada agente se comunicasse com todos os demais.
- Separação entre criação e execução dos sons. Apenas o *outputter* precisa cuidar de seqüenciamento e metrificação das frases geradas pelos músicos.
- Saída centralizada. Simplifica o acesso à placa de som e possibilita gravar um arquivo MIDI central com toda a performance.

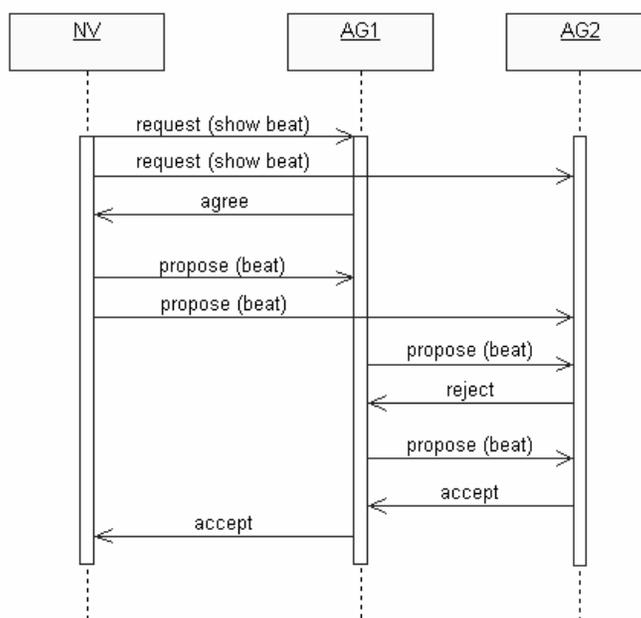
## 5. Agentes Percussionistas

Atualmente, o CInBalada é composto de um único tipo de agente percussionista, chamado **search-agent**. Este nome vem do mecanismo de negociação empregado para escolha dos padrões que serão tocados pelos agentes, que funciona como uma busca intensiva pela melhor combinação de padrões. Para julgar as combinações de padrões, o sistema usa uma função de avaliação rítmica (descrita em detalhes na seção 5.1).

Quanto à **organização** do sistema, os agentes se reúnem em uma estrutura horizontal chamada “Banda”. Apesar de não haver qualquer relação hierárquica entre os agentes, os agentes se comunicam por ordem de entrada na Banda, para facilitar as interações. Ou seja, durante uma rodada de negociação, o agente que primeiro ingressou na Banda toma uma decisão e repassa para o segundo que repassa para o terceiro, etc.

Quando um novo agente é inserido no sistema, ele inicia um processo de **negociação** com os agentes da Banda para tentar encontrar uma combinação entre os padrões rítmicos de todos os agentes de maneira que, dois a dois, todos os padrões “combinem”, segundo a função de avaliação. No decorrer da negociação, o agente novato sucessivamente apresenta cada um de seus padrões rítmicos à Banda (em formato XML). Os agentes da Banda interagem entre si tentando mudar seus padrões de maneira a manter a coerência, tanto internamente como com o agente novato. Se isso não for possível, o agente novato é rejeitado e encerrado.

Apresentamos na figura 4 abaixo um exemplo de uma rodada de negociação. Nesse exemplo, existem três agentes: o agente **NV**, que deseja entrar na Banda, e dois outros agentes, **AG1** e **AG2**, que já fazem parte da Banda, sendo AG1 o mais antigo deles.



**Figura 4. Exemplo de negociação entre os agentes percussionistas.**

A figura acima mostra um exemplo de negociação que terminou em sucesso, ou seja, terminou com a entrada do agente NV na Banda. Esse exemplo começa com o agente NV enviando um pedido para se apresentar à Banda (as mensagens vindas de NV são sempre enviadas a todos). Após ter seu pedido aceito, o agente NV manda uma proposta de batida (padrão rítmico) à Banda. Ao receber a proposta, o agente AG1 analisa seu repertório e escolhe uma batida adequada à que foi apresentada por NV. O agente AG1, então, repassa esse padrão escolhido para AG2 (mensagem “propose”), que a recusa, por não encontrar em seu repertório um padrão rítmico que case com os dois já apresentados (o de NV e o de AG1). O agente AG1 propõe uma outra batida que também casa com a de NV e, dessa vez, o agente AG2 consegue encontrar um padrão que se adeque tanto ao de NV quanto ao de AG1. O agente AG1, representando toda a Banda, comunica a NV que sua batida foi aceita, permitindo-o ingressar na Banda.

## 5.1. Função de Avaliação

O objetivo da função de avaliação no CInBalada é medir a “**qualidade musical**” das combinações de padrões rítmicos escolhidas pelos agentes do sistema. Ela é usada durante o processo de negociação entre os agentes para definir quando um par de padrões é “aceitável”.

A escolha de uma função de avaliação é um dos pontos mais importantes do sistema. No entanto, essa é uma tarefa muito difícil, devido à ausência de trabalhos teóricos sobre esse tema. Para fazer um julgamento mais completo sobre a qualidade musical de um par de padrões tocados em conjunto, é necessário levar em conta vários critérios musicais, tais como: complexidade, paralelismo (auto-similaridade), nível de tensão, etc.

Aparentemente, não existem trabalhos que ofereçam medidas objetivas (funções) sobre a maior parte desses critérios. Portanto, desenvolvemos uma função de avaliação própria para julgamento das combinações de padrões. Essa função baseia-se na noção de que são necessárias tanto uma certa dose de afinidade (ou similaridade) entre os padrões, quanto um pouco de originalidade (diferenças) entre eles, para que soem bem em conjunto. Assim, a função consiste simplesmente em medir as quantidades de pontos de encontros e de pontos de desencontros entre ataques dos dois padrões. O valor da função será o módulo da diferença entre esses valores. Assim, quanto menor o valor, mais equilíbrio entre encontros e desencontros e, portanto, maior a coerência musical entre os dois padrões.

Para comparações com padrões com mais de um compasso (porém, de mesmo tamanho), é feita uma normalização, dividindo o valor da função pelo número de compassos. No caso de comparar padrões de diferentes tamanhos, calculamos o Mínimo Múltiplo Comum entre os tamanhos e concatenamos cada padrão consigo mesmo até que todos tenham esse mesmo tamanho. A partir daí, a comparação pode acontecer normalmente.

## 6. Implementação

O CInBalada está sendo desenvolvido em C++ e possui três módulos centrais: uma biblioteca musical, a plataforma multiagentes e os agentes do sistema. No decorrer dessa seção apresentamos sucintamente cada um dos módulos implementados, bem como os resultados obtidos com o sistema.

### 6.1. Biblioteca Musical

Criada para realizar as manipulações sonoras necessárias para a implementação dos agentes do CInBalada. O principal foco da biblioteca é a tecnologia MIDI. A biblioteca permite manipular arquivos MIDI e enviar eventos MIDI para a saída sonora. Além disso, há um conjunto de classes que mantém as informações de alto nível sobre os padrões rítmicos (mostradas na seção 3.1), armazenando-as no formato XML. Essas funcionalidades estão concentradas na classe `RhythmPattern` que é uma representação de alto-nível de um padrão rítmico de um instrumento de percussão.

## 6.2. Plataforma Multiagente

Para permitir a operação dos vários agentes do sistema foi implementada uma plataforma multiagentes para C++: o FAMA (*Framework* para Aplicações MultiAgentes). O FAMA é inspirado no JADE (Bellifemine, et.al 1999) e foi implementado devido à carência de plataformas multiagentes para C++. A plataforma oferece todos os serviços básicos recomendados pela FIPA: gerenciamento do ciclo de vida dos agentes, serviços de localização de agentes (páginas amarelas e páginas brancas) e transporte mensagens. No entanto, a plataforma do FAMA não é distribuída, uma vez que distribuição não é um requisito do projeto CInBalada.

## 6.3. Agentes do CInBalada

O agente *ouputter* é representado pela classe `OutputterAgent` e é criado automaticamente quando a plataforma FAMA é iniciada. Ao ser criado, ele cadastra o seu serviço como “*MIDI Output Service*” no serviço de páginas amarelas da plataforma e, então, inicia a sua execução, sendo responsável pelas tarefas de: receber requisições de cadastramento de agentes produtores de música e ouvintes, receber frases dos agentes, enviar requisições periódicas de frases e seqüenciar a saída sonora (placa de som e arquivo).

Os agentes percussionistas são instâncias da classe `SearchAgent` e guardam um conjunto de objetos `RhythmPattern` da biblioteca musical, que representam os padrões rítmicos conhecidos pelo agente. Os agentes percussionistas podem possuir apenas um de dois possíveis *comportamentos* (sub-classes de `Behaviour`): `JoinerBehaviour` e `PlayerBehaviour`.

O `JoinerBehaviour` implementa o papel do agente novato no protocolo de negociação dos agentes percussionistas. Inicialmente, ele faz uma requisição para entrar na Banda. Se algum padrão do agente for aceito, o `JoinerBehaviour` é encerrado e um `PlayerBehaviour` é instanciado para o mesmo agente. Se não for aceito, o agente é encerrado. Já a classe `PlayerBehaviour` implementa o comportamento de um agente que faz parte da Banda. Ele envia periodicamente seu padrão rítmico corrente ao *ouputter* sempre que requisitado e, quando surge uma requisição de um agente novato para entrar na Banda, ele inicia a negociação interna (com os demais agentes da Banda) para tentar se adaptar ao novo agente<sup>1</sup>.

A figura 5 a seguir mostra o diagrama de classes da implementação dos agentes, mostrando também as principais relações com classes dos outros dois módulos. As classes mais escuras são as que formam os agentes, enquanto as classes claras são parte da biblioteca musical ou do *FAMA*.

---

<sup>1</sup> Conforme o mecanismo de negociação mostrado na seção 5.

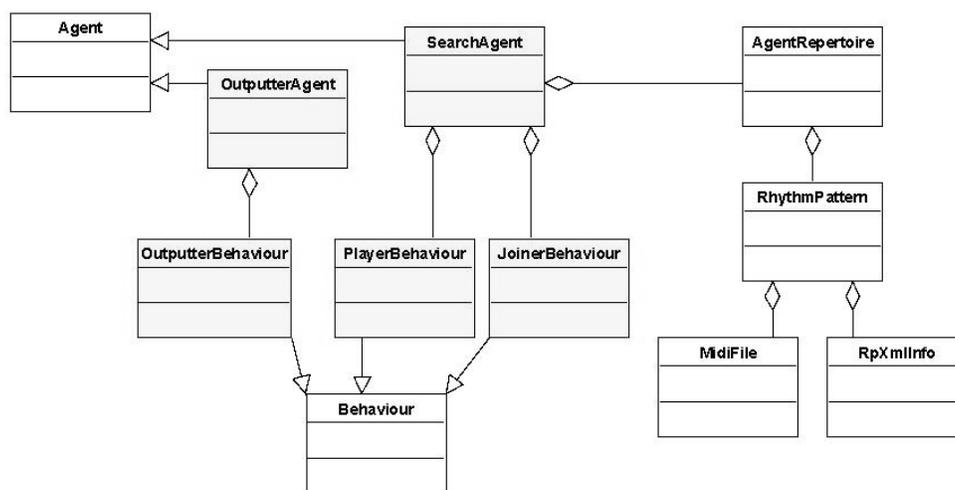


Figura 5. Diagrama de classes dos agentes do CinBalada.

## 6.4. Resultados

Os testes preliminares feitos com a presente versão do sistema mostraram resultados promissores. O sistema permite obter, facilmente, diversos arranjos de percussão com coerência musical entre os agentes aceitável, desde que sejam escolhidos poucos instrumentos (por volta de quatro). Com muitos instrumentos, o protocolo de negociação tende a rejeitar novos agentes e, quando os aceita, gera uma saída musical confusa, porque não há qualquer organização dos instrumentos dentro da performance.

## 7. Conclusão e Trabalhos Futuros

Neste artigo apresentamos o CinBalada, um sistema musical inteligente que permite a criação de arranjos puramente rítmicos variados. Para isso, o sistema dispõe de agentes percussionistas capazes de interagir e negociar entre si para manter a performance musicalmente aceitável, segundo uma função de avaliação interna. Por meio de uma interface gráfica, usuários humanos podem escolher diferentes formações e atribuir diferentes estilos a cada agente, ouvindo o resultado imediatamente. Desse modo, o CinBalada funciona como um laboratório de ritmos onde diferentes misturas podem ser experimentadas.

O CinBalada traz, basicamente, duas inovações em relação aos sistemas de Computação Musical atuais: a criação semi-automática de arranjos rítmicos polifônicos (múltiplos instrumentos) e a implementação de mecanismos de interação entre os agentes mais próximos do que acontece no mundo real, o que representa um passo importante para a criação de um sistema interativo no futuro. No entanto, no desenvolvimento do CinBalada ainda existem questões em aberto, que pretendemos abordar em um futuro próximo. Abaixo discutimos algumas dessas questões.

Em primeiro lugar, é muito importante pesquisar funções de avaliação que tenham significado musical claro, para melhorar a coerência das saídas musicais (e.g. medidas de complexidade, medidas de similaridade). Talvez seja deixado a critério do usuário escolher quais funções usar, ou qual a importância relativa de cada função de avaliação, para que as saídas do sistema se moldem ao gosto do usuário. Uma outra ex-

tensão interessante é a possibilidade de os agentes assumirem diversos papéis na Banda (e.g. solista, executor de virada e regente), de modo a organizar a função de cada instrumento dentro da performance. Esses papéis seriam escolhidos pelos usuários e dariam aos agentes diferentes graus de liberdade para a execução de suas performances, enriquecendo o realismo da simulação provida pelo CInBalada.

Também pretendemos enriquecer a interface com o usuário de modo a oferecer mais opções de personalização das performances. Porém essa é uma tarefa que ocorrerá a medida que novas características forem incorporadas ao sistema. Queremos, ainda, enriquecer a base de padrões rítmicos do sistema de modo a comportar uma maior variedade de instrumentos e de estilos musicais. Pretendemos aproveitar a riqueza rítmica da música brasileira adicionando padrões de estilos musicais típicos do Brasil, tais como: samba, baião e maracatu.

Por fim, estamos cientes da necessidade de realizar mais experimentos com o sistema e, por isso, planejamos a realização de experimentos com especialistas (músicos e/ou professores de música) no decorrer do desenvolvimento das próximas extensões. Eles terão a oportunidade de testar a ferramenta e, por meio de um questionário, julgá-la segundo critérios musicais e funcionais.

## 8. Referências

- Bellifemine, F. Rimassa, G. Poggi, A. (1999). JADE - A FIPA-compliant Agent Framework. Em: *Proceedings of the 4th International Conference and Exhibition on The Practical Application of Intelligent Agents and Multi-Agents*. Londres.
- Ganascia, J.G. Ramalho, G. Rolland, P.Y. (1999). An Artificially Intelligent Jazz Performer. *Journal of New Music Research*, 28 (2), pp. 105-129. Amsterdã: Swets & Zeitlinger.
- FIPA: Foundation for Intelligent Physical Agents. (2003). Informações no site: <http://www.fipa.org/> (acesso em 16 de junho de 2005).
- Gannon, Peter. (2005). *Band-in-a-Box* [Programa de Computador]. Hamilton, Ontário: PG Music Inc.
- Honning, H. (2002). Structure and Interpretation of Rhythm and Timing. *Tijdschrift voor Muziektheorie* [Jornal de Teoria Musical], 7 (3), pp. 227-232.
- Kragtwijk, M. Nijholt, A. Zwiers, J. (2001). Implementation of a 3D Virtual Drummer. Em: *Computer Animation and Simulation 2001*, Editores: Magnenat-Thalmann, N., Thalmann, D. Viena: Springer Verlag Wien.
- Ladzekpo, C. K. (n.d.). *Rhythmic Principles*. Disponível em: <http://cnmat.cnmat.berkeley.edu/%7Eladzekpo/PrinciplesFr.html> (acesso 13 de junho de 2005).
- Lerdahl, F. Jackendoff, R. (1983). *A Generative Theory of Tonal Music*. MIT Press.
- Messick, Paul (1998). *Maximum MIDI: Music Applications in C++*. (S.l.): Manning Publications.
- Miranda, E. R. (2002). Emergent sound repertoires in virtual societies. *Computer Music Journal*, 26(2), pp. 77-90.

- Murray-Rust, D. (2003). *VirtuaLatin - Agent Based Percussive Accompaniment*. Dissertação (MSc in Informatics), School of Informatics, University of Edinburgh, Edimburgo.
- Papadopoulos, G. Wiggins, G. (1999). AI Methods for Algorithmic Composition: A Survey, a Critical View and Future Prospects. Em: *Proceedings of the AISB'99 Symposium on Musical Creativity*, pp. 110-117. Brighton, UK: SSAISB.
- Pachet, F. (2000). Rhythms as Emerging Structures. Em: *Proceedings of the 2000 International Computer Music Conference (ICMC)*. Berlim : ICMA.
- Pachet, F. (2002). Interacting with a Musical Learning System: The Continuator. Em: *Proceedings of the II International Conference on Music and Artificial Intelligence (ICMAI)*. Edimburgo.
- Pearce, M. (2000). *Generating Rhythmic Patterns: a Combined Neural and Evolutionary Approach*. Dissertação (MSc in Artificial Inteligence), School of Informatics, University of Edinburgh, Edimburgo.
- Salazar, M. (1991). *Batucadas de Samba*. Rio de Janeiro : Lumiar Editora.
- Wooldridge, M. J. (2002). *An Introduction to Multi-Agent Systems*. (S.l.): John Wiley and Sons.
- Wulfhorst, R. D. Nakayama, L. Vicari, R. M. (2003). A Multiagent Approach for Musical Interactive Systems. Em: *Proceedings of the Second International Joint Conference on Autonomous Agents and Multiagent Systems*, pp. 584-591. Melbourne: ACM Press.