



UNIVERSIDADE DE SÃO PAULO
PRÓ-REITORIA DE GRADUAÇÃO
CURSO DE CIÊNCIAS MOLECULARES

PROJETO DE INICIAÇÃO CIENTÍFICA

Interfaces entre o Usuário e Sistemas Multiagentes Musicais

Autor:
Pedro H. R. BRUEL
Email:
pedro.bruel@gmail.com

Telefone: 97587-2511

Turma: 20

Orientador:
Prof Dr Marcelo QUEIROZ

Email:
mqz.mqz@gmail.com

Unidade:
Instituto de Matemática e Estatística

Departamento:
Ciência da Computação

1º Semestre de 2013

Sumário

1	Introdução	3
1.1	Sistemas Multiagentes	3
1.1.1	Agente	3
1.1.2	Ambiente	3
1.1.3	Sistemas Multiagentes Musicais	4
1.2	Objetivo	4
2	Fundamentação Teórica	4
2.1	Aplicações de Sistemas Musicais Multiagentes	4
2.1.1	Simulação	5
2.1.2	Geração	5
2.1.3	Reação à Performance	5
2.1.4	Geração e Improvisação	5
2.2	Exemplos	5
2.2.1	CAMUS e Chaosynth	5
2.2.2	An Artificially Intelligent Jazz Performer	6
2.2.3	Living Melodies	6
2.2.4	Projeto Andante	7
2.3	Ensemble	8
2.3.1	Arquitetura	8
2.3.2	Interfaces	8
2.3.3	Código e Documentação	9
2.4	Pure Data	9
2.4.1	libpd	9
3	Metodologia	9
4	Cronograma	10
4.1	Desenvolvimento do Projeto	10
4.2	Disciplinas	10
5	Conclusão	10

Resumo

A adaptabilidade em tempo real e o potencial de modelagem de sistemas complexos são características de Sistemas Multiagentes, que têm variadas aplicações no contexto musical. A ferramenta de criação musical Pure Data (Pd) é tradicionalmente utilizada por músicos em preparações musicais, e o objetivo deste projeto é produzir uma integração do arcabouço para Sistemas Multiagentes Musicais Ensemble - produzido pelo Grupo de Computação Musical do IME/USP - com a libpd, uma infraestrutura do tipo API para Pure Data. A integração deve facilitar a utilização do arcabouço por usuários não-programadores. As funções que integrarão os dois sistemas devem mapear as estruturas de dados correspondentes, gerenciando o fluxo de informação entre os dois softwares.

1 Introdução

A utilização de algoritmos computacionais no processo de composição musical é parte de uma tendência que busca valorizar o dinamismo em processos artísticos de criação [23], explorando a tecnologia de Agentes [26] junto a conceitos de vida artificial [19].

O Sistema Ensemble [23] busca oferecer flexibilidade e generalidade na especificação de aplicações criativas de sistemas multiagentes, mas essa mesma flexibilidade e generalidade faz com que ele apresente uma curva de aprendizado acentuada aos compositores interessados, um problema recorrente na área [26]. Este projeto prevê uma integração do Ensemble com o Pure Data, que é uma ferramenta bastante popular entre músicos interessados em tecnologia, e espera-se que tal integração irá suavizar substancialmente a curva de aprendizado do sistema Ensemble.

1.1 Sistemas Multiagentes

Um Sistema Multiagentes é composto [27, 15] de um Ambiente virtual e Agentes computacionais. A interação entre os Agentes contidos no ambiente é influenciada pelos parâmetros e características desse Ambiente. As seções seguintes detalharão as propriedades de cada componente.

1.1.1 Agente

Entidade computacional com interfaces e características bem definidas, orientada e instrumentalizada para a solução de problemas computacionais em geral [15]. Para o propósito deste projeto, significa uma estrutura que apresenta as seguintes propriedades [27]:

1. **Autonomia:** Encapsula estados e é capaz de tomar decisões independentes, baseadas nesses estados;
2. **Reatividade:** Está imerso em um Ambiente, é capaz de percebê-lo e responder a mudanças;
3. **Proatividade:** Pode apresentar ações orientadas por objetivos, tomando iniciativas para alcançá-los;
4. **Interação Social:** É capaz de interagir com outros Agentes, ou com usuários, para resolver problemas cooperativamente e atingir seus objetivos.

1.1.2 Ambiente

Contém os Agentes e fornece um meio para as interações entre eles. Pode ser implementado como uma simulação do mundo físico, uma interface gráfica com o usuário, uma coleção de outros Agentes, e variações e combinações dentre essas possibilidades [27].

1.1.3 Sistemas Multiagentes Musicais

Um Agente Musical é um Agente especializado em processamento e síntese de sinais sonoros e informação musical, e é tipicamente capaz de receber informação sonora proveniente do Ambiente ou de outros Agentes através de seus Sensores, aplicar algum tipo de Raciocínio sobre essa informação e responder de forma adequada através de Atuadores, levando em consideração seus objetivos e Base de Conhecimentos - onde ficam armazenadas as informações necessárias ao seu funciona-

mento [24, 22, 23].

No contexto de um Sistema Multiagentes Musical, um Ambiente é composto de uma Representação Física - incluindo definições de parâmetros de propagação de som, dimensões espaciais e temporais, e todas as grandezas físicas relevantes à representação sonora - e uma Representação Ecológica, que define as regras para interação de Agentes, como tempo de vida, reprodução, movimentação e demais funções biológicas ou comportamentais [24, 22, 23].

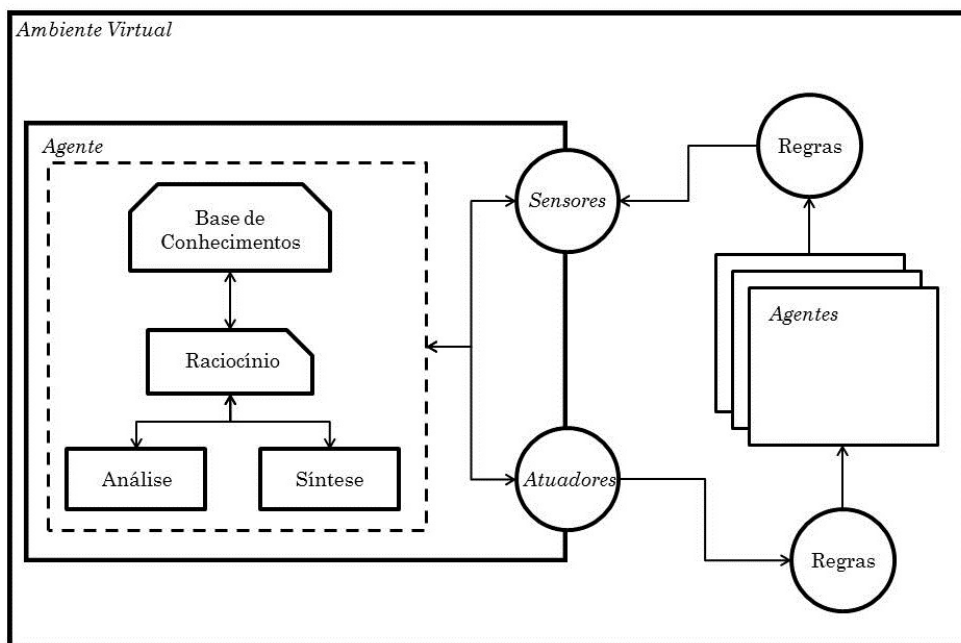


Figura 1: Representação de um Sistema Multiagentes Musical

1.2 Objetivo

O objetivo deste projeto é construir uma biblioteca de funções que produzam um mapeamento conceitual e integração estrutural do Ensemble com a linguagem Pure Data, criando novos componentes em C que utilizem a libpd, bem como *externals* para Pd (escritos em C/C++).

Essa integração tem como objetivo facilitar a interação do usuário com as funcionalidades do arcabouço, tendo em mente especial-

mente o usuário familiar com Pure Data sem experiência em Java.

2 Fundamentação Teórica

2.1 Aplicações de Sistemas Musicais Multiagentes

Whalley [26], apresenta quatro abordagens para o uso de Sistemas Multiagentes no contexto artístico de música e som.

2.1.1 Simulação

Agentes simulam estilos pré concebidos, lidando com situações multi-causais em tempo real sem entrada humana, interagindo uns com os outros na construção de uma estrutura musical.

Em Ramalho *et. al* [21], é apresentado um modelo para um baixista virtual numa banda de jazz, numa apresentação ao vivo, sujeito a uma platéia e demais músicos.

Em Costalonga *et. al* [12], o modelo é para a interação entre as mãos de um guitarrista, que decidem formatos de acordes no braço da guitarra.

2.1.2 Geração

Neste tipo de aplicação o foco artístico é o processo de criação, em detrimento de um produto com forma pré-determinada. O compositor cria uma série de comportamentos [19] para os Agentes, e a geração de conteúdo e até mesmo de uma estrutura musical é resultado da interação entre eles.

2.1.3 Reação à Performance

Agentes têm seus comportamentos e ações pré-programados para interagir com usuários, criando variados tipos de interação em redes com entrada humana.

2.1.4 Geração e Improvisação

Permite entrada de improvisações humanas mas, ao invés de uma simples reação por parte dos Agentes, ocorrerá geração adaptativa de uma resposta. A resposta será então processada pelo improvisador humano, que gerará nova entrada.

2.2 Exemplos

Nesta seção, farei a exposição de alguns Sistemas Musicais Multiagentes apresentados por diferentes artigos. Os exemplos apresentados procuram representar os contextos descritos anteriormente.

2.2.1 CAMUS e Chaosynth

O artigo [18] de Eduardo Miranda trata do uso de processos computacionais em composição musical, e apresenta o CAMUS e o Chaosynth, dois sistemas musicais baseados em autômatos celulares.

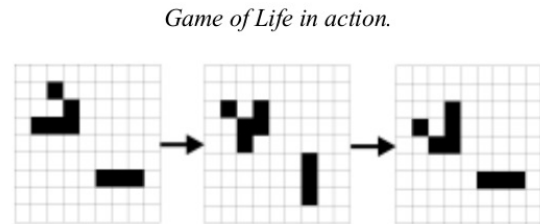


Figura 2: Iterações do Tabuleiro do Game of Life [18]

Uma das partes do CAMUS é uma implementação do Game of Life, um tabuleiro bidimensional de autômatos celulares, modelando uma colônia de organismos virtuais que seguem regras simples. O tabuleiro é então iterado simulando interações entre esses organismos, com cada passo representando um estado diferente do sistema.

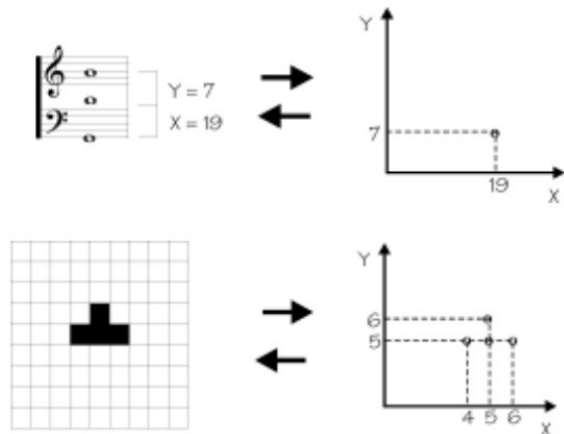


Figura 3: Correspondência Musical do Tabuleiro [18]

Uma célula preta representa um organismo vivo, e uma branca, um local vago. O estado de uma célula numa iteração dependerá das células adjacentes nos instantes anteriores.

Cada célula representa uma tripla de notas, e cada passo do Game of Life gera diferentes triplas.

O segundo Sistema Musical Multiagente apresentado pelo artigo é o Chaosynth, que funciona essencialmente como um sintetizador granular, onde cada grânulo é representado por um autômato celular de três estados, e as interações entre esses autômatos dão origem aos eventos sonoros.

Os estados possíveis são quiescente, despolarizado ou queimado, e se relacionam com valores de "corrente elétrica". Esses valores dependem do estado anterior de determinada célula, bem como do estado de seus vizinhos. Basicamente, uma célula quiescente começa a se despolarizar, e ao atingir um limite de despolarização, dispara e queima. Uma célula queimada num instante t é substituída por uma quiescente no instante $t + 1$.

Uma distribuição aleatória de células na grade do Chaosynth tende a um ciclo de padrões oscilatórios, e a soma dos sons produzidos pelos grânulos representados nas células forma eventos sonoros maiores.

The CA used in Chaosynth tends to evolve from (a) an initial random distribution of cells in the grid (b) towards an oscillatory cycle of patterns.

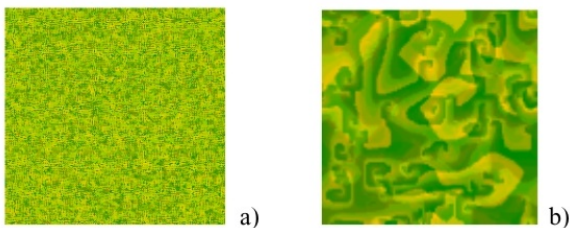


Figura 4: Evolução de Padrões no Chaosynth [18]

O material produzido nas iterações do Game of Life e do Chaosynth é utilizado nos passos seguintes do CAMUS, e o resultado é um material musical utilizado nas composições dos autores. Exemplos de composições utilizando material desses sistemas são *Entre l'absurde et le mystere - for chamber orchestra*,

Wee Batucada Scotica e Olivine Trees.

2.2.2 An Artificially Intelligent Jazz Performer

Neste artigo [21] é apresentado um Sistema Musical Multiagente que um consiste em um modelo para simulação de um baixista de uma banda de Jazz, que interage com outros músicos durante uma performance ao vivo.

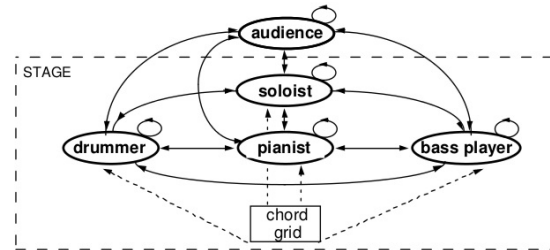


Figura 5: Mapa de Interações do Agente Musical [21]

Na performance de Jazz, existe uma grande diferença entre as instruções da partitura e o que está sendo tocado pelos músicos. Para lidar com esta diferença, o Agente Musical deve armazenar fragmentos melódicos previamente conhecidos ou escutados em sua memória musical, e deve também ter implementadas em seu raciocínio regras para o uso desses fragmentos no contexto musical mutável da performance [21].

2.2.3 Living Melodies

É um Sistema Musical Multiagente com características emergentes. Os agentes têm comportamentos codificados em instruções, que podem se modificar ao longo da execução do programa, dado o contato com outros agentes e com o ambiente [13].

Assim como no CAMUS e Chaosynth, os organismos virtuais são representados em uma grade, onde cada célula contém um organismo. Mas neste Sistema os organismos simulados são muito mais complexos, pois têm codificado em sua Base de Conhecimentos uma série de instruções comportamentais que podem

ser mutadas e transmitidas a novos organismos, no processo de reprodução. Esses Agentes podem se mover pelo Ambiente, escutar som produzido por outros Agentes e interagir com eles.

R	rest for one time step
W	walk one step in the current direction
S	sing a note
T-	turn right
T+	turn left
Ts	turn toward sound if listening pleasure is high enough (>1)
L(loop-count){number of instructions included}	a loop instruction
{ < or = or > }{sensor}{sensor or constant}	an IF clause
(number of instructions to include)	
a	right tentacle, returns the species no. of the creature in the square in the right-forward direction
b	front tentacle, see above
c	left tentacle, see above
d	listening pleasure value
e	sound volume at the current coordinates
f	sound direction at the current coordinates
g	age of the creature itself
h	direction of the creature
00-99	integer constants for comparisons

Figura 6: Instruções e Parâmetros Constituintes do Genoma [13]

O comportamento das diferentes espécies de Agente definidas no Living Melodies é determinado por seu genoma procedural e sonoro. O genoma procedural define uma sequência de ações a ser realizada pelo Agente e também seu mecanismo de produção sonora, e o genoma sonoro contém as notas musicais favoritas desse Agente.

WorldSize = 100	World size
SoundRadius = 20	How far the sound can be heard
MaxSoundGenomeLength = 5	Max size of the list of preferred notes
MaxActionGenomeLength = 12	Max length of the initial genomes
MaxAge = 4000	Age limit
MinLifePoints = 1	Life-points needed to stay alive
MaxNumberOfCreatures = 10000	Max number of creatures
MaxLoopLength = 5	In the action genome
MaxLoopSteps = 5	In the action genome
MaxIfSteps = 5	In the action genome
InitCreatureProb = 0.01	Initial population density
InitFoodProb = 0.05	Initial food density
CostOfSinging = 0.1	Cost of singing
CostOfTime = 0.1	Cost of existence per unit time
SoundCrossoverProb = 0.2	Crossover probability per genome
ActionCrossoverProb = 0.1	Crossover probability per genome
MutateSpecies = 0.05	Mutation probability
MutateRandomCreature = 0.02	New Random Creature
MateSpread = 5	Dispersion radius for offspring and one parent
MustWalkInterval = 5	Mating criterion
DeathLength = 30	If genome gets bigger—die!

Figura 7: Parâmetros da Simulação [13]

Os Agentes podem se reproduzir desde que atinjam alguns criterios definidos na implementação, como quantidade de energia, quantidade de felicidade - obtida ao se ouvir no-

tas preferidas, idade e proximidade de outro Agente.

Há uma implementação do Living Melodies utilizando o Arcabouço Ensemble, e um passo da execução é mostrado na Figura X.



Figura 8: Living Melodies no Ensemble [23]

O som produzido pelo Living Melodies dependerá do que for emitido pelos Agentes contidos na simulação, além da quantidade de felicidade dos Agentes emissores e de outros parâmetros definidos, como por exemplo a localização do ouvinte no Ambiente virtual.

2.2.4 Projeto Andante

O Projeto Andante consiste em uma infraestrutura de código aberto para a construção de aplicações distribuídas para composição e performance musical baseadas em agentes móveis musicais [25].

Um Agente capaz de se migrar entre máquinas conectadas por uma rede é chamado Agente Móvel. Este tipo de Agente tem autonomia, dentro do Sistema, para decidir o que fazer na máquina em que se encontra e quando migrar para outra máquina.

O conceito de Agente Móvel surgiu na década de 90 [16], apresentando um novo pa-

radigma para sistemas computacionais distribuídos e móveis.

2.3 Ensemble

O Ensemble é um arcabouço para a construção de Sistemas Musicais Multiagentes, descrito por Thomaz [23] e Santiago [22], e tem como público alvo a comunidade de músicos que utiliza o computador como ferramenta de criação musical, fornecendo os serviços necessários para a criação de Sistemas Musicais Multiagentes. O usuário pode então programar as funções específicas para seus objetivos musicais, apenas parametrizando componentes e agentes já prontos, ou programando novos.

O controle do fluxo de execução é mantido pelo código interno e não por métodos externos implementados pelo usuário, que pode acrescentar novos componentes aos Agentes em tempo de execução do sistema.

A programação de novos componentes para o Ensemble requer conhecimento de seu funcionamento, e deve seguir as convenções de programação do arcabouço. Nesta situação, segundo Johnson e Foote [17], ele é classificado como um arcabouço do tipo *white-box*. Na mesma nomenclatura, uma aplicação que utilizasse apenas componentes reutilizáveis do Ensemble o caracterizaria como *black-box* como black-box, pois poderia ser construída sem conhecimento do funcionamento interno do arcabouço.

2.3.1 Arquitetura

A arquitetura [23] do Ensemble segue o paradigma de programação orientada a objetos, exigindo uma linguagem capaz de implementá-lo.

A presente implementação do Ensemble foi codificada em linguagem Java, a fim de permitir que aplicações musicais possam ser executadas em plataformas distintas.

A arquitetura utiliza o middleware para sistemas multiagentes JADE [11], que provê a infra-estrutura necessária para manter o ci-

clo de vida dos Agentes e controlar a troca de mensagens entre eles.

Principalmente, três funcionalidades do JADE são utilizadas na implementação do Ensemble, o serviço de agendamento de tarefas, o mecanismo de troca de mensagem entre os agentes e o serviço de diretório.

2.3.2 Interfaces

O arcabouço Ensemble é capaz de interagir com sistemas externos, usuários ou bibliotecas de processamento de áudio, e tem suporte a vários protocolos de comunicação. Essas interfaces permitem incorporar recursos de processamento sonoro ou ferramentas para criação e gerenciamento de Agentes, que provenientes de outras bibliotecas e enriquecem as funcionalidades do arcabouço.

Uma interface gráfica, baseada no Agente *Sniffer* disponível no middleware JADE [11], foi concebida para controlar e monitorar visualmente o sistema.

Essa interface permite o envio de mensagens customizadas para um determinado componente do sistema, a criação e destruição de Agentes musicais, e a inserção e remoção de componentes e Ambientes. Além disso, é possível visualizar todos os Agentes, componentes registrados e seus estados internos, e alterar qualquer um de seus parâmetros [23].

O arcabouço Ensemble é capaz de se comunicar via protocolo Open Sound Control, podendo receber mensagens enviadas por *patches* Pure Data e enviar suas próprias mensagens [23].

Essa comunicação permite a um *patch* Pd ter acesso a estados e parâmetros do Ambiente Ensemble e de Agentes no arcabouço.

O protocolo de comunicação Open Sound Control é uma sintaxe que permite troca de mensagens entre computadores, sintetizadores e softwares, possui um sistema de nomenclatura capaz de endereçamento de mensagens a múltiplos destinatários, mecanismos de pacotes de mensagens (*bundles*) e controles temporais de alta resolução (*time tags*) [28].

O Ensemble tem incorporadas duas bibliotecas Java para processamento digital de sinais, a `aubio` [1] e a `LibXtract` [8].

Para interfaces de entrada/saída de áudio, foram implementados o `JACK` [5], o `PortAudio` [9] e o `JavaSound` [7], todos através da `JNI` (Java Native Interface) [6].

É possível incorporar novas bibliotecas de processamento sonoro ou de entrada/saída de áudio através de código Java, caso essas bibliotecas tenham sido implementadas nessa linguagem, ou através de código nativo com a `JNI`.

2.3.3 Código e Documentação

O código e a documentação do Arcabouço Ensemble podem ser encontrados respectivamente em [3] e [4]. O projeto tem uma página [2] no site do Grupo de Computação Musical do IME, e o código pode ser obtido seguindo as instruções na própria página onde está hospedado.

2.4 Pure Data

A linguagem de programação visual Pure Data, ou `Pd` [20], é uma ferramenta tradicionalmente utilizada por músicos no processo de criação e composição de material musical [23].

Trabalhando com entradas (*inlets*) e saídas (*outlets*) ligadas entre si através de *patches*, é possível estruturar blocos de processamento e manter fluxos de áudio ao longo do tempo.

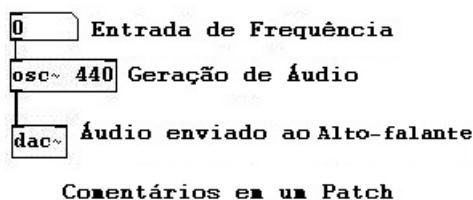


Figura 9: Um *patch* muito simples em `Pd` [14]

Diferentemente de Agentes em um Sistema Multiagentes Musical, os blocos de processamento em Pure Data trabalham de forma aco-

plada e determinada, seguindo uma estrutura rígida de fluxo sonoro [23].

A criação de aplicações musicais e de processamento sonoro mais complexas é feita a partir da combinação de componentes visuais simples, chamados **objetos**. Diversas funcionalidades relacionadas ao processamento digital de sinais e à manipulação de entrada e saída já estão implementadas nas distribuições do `Pd`, e estão prontas para serem conectadas a outros objetos no ambiente gráfico.

É possível criar e incorporar novas ferramentas à linguagem utilizando código escrito em `C/C++`, que será compilado como biblioteca dinâmica e incorporado ao ambiente `Pd`, como objeto. O objeto criado no processo de estender o `Pd` com código `C/C++` é chamado de **external**.

2.4.1 libpd

É uma biblioteca que integra patches e funcionalidades do `Pd` a diversas linguagens, permitindo escrever código que será executado independentemente, ou gerar novas bibliotecas para integração e uso do Pure Data dentro de outros softwares [10].

3 Metodologia

Inicialmente, foi feito um estudo aprofundado da literatura referente aos conceitos abordados pelo projeto. A documentação e código do arcabouço Ensemble, do Pure Data e da `libpd` também foram e continuarão serão estudados.

Após as revisões da literatura e estudos de código e documentação, a interface entre os softwares deverá ser implementada, em paralelo aos testes objetivos de desempenho e subjetivos de interatividade com o usuário.

Um exemplo de integração entre o arcabouço Ensemble, o `Pd` e o protocolo `OSC` é a possibilidade futura de definir o raciocínio interno de um Agente, que hoje é escrito em Java, como um *patch* Pure Data. Os Sensores se comunicariam com *inlets*, direcionando a

informação sonora do Ensemble para ser processada pela libpd - e os resultados desses processamentos seriam transferidos para o Ambiente virtual do arcabouço através da comunicação de *outlets* com Atuadores.

Serão escritos externals para o Pd que farão a correspondência conceitual dos Agentes no Ensemble para objetos no ambiente Pure Data, e utilizarão funções da implementação em C do Pure Data para comunicação com a interface gráfica, o processamento e a entrada/saída de áudio. Esses externals serão então incorporados ao arcabouço através da Java Native Interface [6].

As Análises de Desempenho previstas no Cronograma visam garantir o tempo de resposta adequado às aplicações musicais, e serão realizadas em paralelo à implementação das interfaces descritas.

4 Cronograma

4.1 Desenvolvimento do Projeto

As etapas descritas na Metodologia serão executadas nos semestres seguintes, procurando seguir a estrutura da Tabela 1.

4.2 Disciplinas

As disciplinas apresentadas na proposta de grade curricular irão contribuir fortemente para a formação do aluno tanto em disciplinas fundamentais da Ciência da Computação, como Estruturas de Dados e Programação Orientada a Objetos, quanto em disciplinas da Música, como Acústica e Música Eletroacústica, além de disciplinas específicas da fronteira interdisciplinar entre Ciência da Computação e Música, como Computação Musical e Processamento de Áudio, Imagem e Vídeo.

	1º Semestre	2º Semestre	3º Semestre
Revisão da Literatura	X		
Estudos de Documentação e Código	X	X	
Implementação da Interface		X	X
Análises de Desempenho		X	X
Experimentos de Interatividade			X

Tabela 1: Desenvolvimento do Projeto

5 Conclusão

Este projeto de pesquisa descreveu sucintamente as propostas de estudo e integração do arcabouço Ensemble para Sistemas Musicais Multiagentes e do ambiente de processamento sonoro Pure Data (Pd). Espera-se que tal integração permita a aproximação de usuários potencialmente interessados na exploração artística dos conceitos de agentes computacionais, atualmente dificultada pela exigência de conhecimentos de programação em Java para a criação de novas estruturas no Ensemble.

Embora o Pd também seja um ambiente computacional, seu uso por músicos interessados em processos eletroacústicos/computacionais é bastante amplo, o que o tornaria uma via de acesso natural ao arcabouço Ensemble.

Não menos importante do que a contribuição específica definida nos objetivos deste projeto, a estrutura curricular multi disciplinar e a participação do aluno nas reuniões coletivas e seminários semanais do grupo de pesquisa em Computação Musical do IME/USP serão de grande importância na formação do aluno e em sua preparação para a pesquisa.

Referências

- [1] Aubio: About. <http://aubio.org/>. Acessado em 15/05/2013.
- [2] Ensemble. <http://compmus.ime.usp.br/ensemble>. Acessado em 18/06/2013.
- [3] Ensemble: Código fonte. <https://code.google.com/p/musicalagents/source/browse/>. Acessado em 18/06/2013.
- [4] Ensemble: Documentação. <http://compmus.ime.usp.br/ensemble/doc/>. Acessado em 18/06/2013.
- [5] Jack. <http://jackaudio.org/>. Acessado em 15/05/2013.
- [6] Java native interface. <http://docs.oracle.com/javase/6/docs/technotes/guides/jni/>. Acessado em 15/05/2013.
- [7] Java sound. <http://www.oracle.com/technetwork/java/index-jsp-140234.html>. Acessado em 15/05/2013.
- [8] Libxtract: About. <http://libxtract.sourceforge.net/>. Acessado em 15/05/2013.
- [9] Portaudio. <http://www.portaudio.com/>. Acessado em 15/05/2013.
- [10] Wiki libpd. <https://github.com/libpd/libpd/wiki>. Acessado em 03/12/2012.
- [11] F. Bellifemine, A. Poggi, and G. Rimassa. Jade—a fipa-compliant agent framework. In *Proceedings of PAAM*, volume 99, page 33. London, 1999.
- [12] L.L. Costalonga, R.M. Vicari, and E.M. Miletto. Agent-based guitar performance simulation. *Journal of the Brazilian Computer Society*, 14(3):19–29, 2008.
- [13] Palle Dahlstedt and Mats G Nordahl. Living melodies: Coevolution of sonic communication. *Leonardo*, 34(3):243–248, 2001.
- [14] Kreidler J. *Programming Electronic Music in Pd*. Wolke Publishing House, 2012.
- [15] N.R. Jennings. On agent-based software engineering. *Artificial intelligence*, 117(2):277–296, 2000.
- [16] Dag Johansen, Robbert Van Renesse, and Fred B Schneider. Operating system support for mobile agents. In *Hot Topics in Operating Systems, 1995.(HotOS-V), Proceedings., Fifth Workshop on*, pages 42–45. IEEE, 1995.
- [17] R.E. Johnson and B. Foote. Designing reusable classes. *Journal of object-oriented programming*, 1(2):22–35, 1988.
- [18] Eduardo Reck Miranda. On the music of emergent behavior: what can evolutionary computation bring to the musician? *Leonardo*, 36(1):55–59, 2003.
- [19] E.R. Miranda and P.M. Todd. A-life and musical composition: A brief survey. In *Proceedings of the IX Brazilian Symposium on Computer Music, Campinas,(Brazil)*. Citeseer, 2003.

- [20] M. Puckette et al. Pure data: another integrated computer music environment. *Proceedings of the Second Intercollege Computer Music Concerts*, pages 37–41, 1996.
- [21] G.L. Ramalho, P.Y. Rolland, and J.G. Ganascia. An artificially intelligent jazz performer. *Journal of New Music Research*, 28(2):105–129, 1999.
- [22] Benavides S.D.D. Raciocínio de agentes musicais composição algorítmica vida artificial e interatividade em sistemas multiagentes musicais. Master’s thesis, USP, 2012.
- [23] L.F. Thomaz. *Um arcabouço para construção de sistemas multiagente musicais*. PhD thesis, Universidade de São Paulo, 2011.
- [24] L.F. Thomaz and M. Queiroz. A framework for musical multiagent systems. *Proceedings of the SMC*, pages 119–124, 2009.
- [25] Leo Kazuhiro Ueda and Fabio Kon. Andante-composição e performance musical utilizando agentes móveis. 2003.
- [26] I. Whalley. Software agents and creating music/sound art: Frames, directions, and where to from here. In *Proceedings of International Computer Music Conference*, pages 691–5, 2005.
- [27] M. Wooldridge and P. Ciancarini. Agent-oriented software engineering: the state of the art. *Lecture notes in computer science*, pages 1–28, 2001.
- [28] M. Wright and A. Freed. Open sound control: A new protocol for communicating with sound synthesizers. In *Proceedings of the 1997 International Computer Music Conference*, pages 101–104. International Computer Music Association San Francisco, 1997.