

Usando Pure Data para criar Sistemas Musicais Multiagentes no Arcabouço Ensemble

Pedro Bruel

pedro.brue1@gmail.com

Grupo de Computação Musical IME/USP

7 de abril de 2014

Estrutura

- 1 Pure Data
- 2 Sistemas Musicais Multiagentes
- 3 Arcabouço Ensemble
- 4 Interface com o Pure Data
- 5 Referências

Pure Data

Linguagem de programação visual, voltada a aplicações interativas musicais e multimídia.

Pure Data

Linguagem de programação visual, voltada a aplicações interativas musicais e multimídia.

Projeto de código aberto, sob licença BSD.

Basic_Elements

A small rectangular box with a light gray background and a thin black border. The word "float" is written in a monospaced font inside the box.

object box



empty object box

A small rectangular box with a light gray background and a thin black border. The word "message" is written in a monospaced font inside the box.

message box (click)

A small rectangular box with a light gray background and a thin black border. The number "0" is written in a monospaced font inside the box.

number box (click and drag/type)

A small rectangular box with a light gray background and a thin black border. The word "symbol" is written in a monospaced font inside the box.

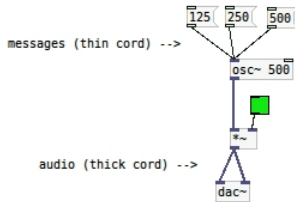
symbol box (click and type)

comment

comment (like the rest of this text)

audio_data

For audio, pd has a specific data type. Audio objects all have a trailing tilde ("~"). Audio coords are thick.



Ambiente Pure Data



Para Saber Mais

Pure Data:

- Manual **FLOSS**: flossmanuals.net/pure-data/
- Código Fonte:

```
$ git clone git://git.code.sf.net/p/pure-data/pure-data
```



Puckette, Miller and others

Pure Data: another integrated computer music environment

Proceedings of the Second Intercollege Computer Music Concerts 37–41

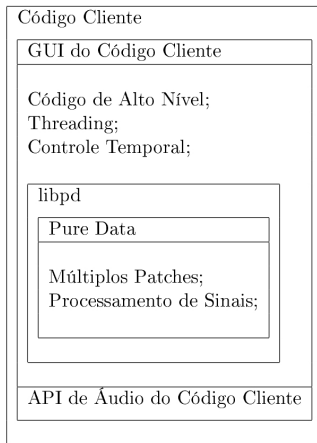


Puckette, Miller (1988)

The Patcher

The-International

Encapsulando o Pure Data



Implementação

Tarefas da Aplicação:

Implementação

Tarefas da Aplicação:

- Inicializar o Pd;

Implementação

Tarefas da Aplicação:

- Inicializar o Pd;
- Definir funções de callback;

Implementação

Tarefas da Aplicação:

- Inicializar o Pd;
- Definir funções de callback;
- Chamadas ao método de processamento;

Implementação

Benefícios:

Implementação

Benefícios:

- Pd em Java, C, Python...;

Implementação

Benefícios:

- Pd em Java, C, Python...;
- Biblioteca DSP;

Implementação

Benefícios:

- Pd em Java, C, Python...;
- Biblioteca DSP;
- Geração de Áudio;

Implementação

Benefícios:

- Pd em Java, C, Python...;
- Biblioteca DSP;
- Geração de Áudio;
- **Configuração e Controle de Aplicações;**

Para Saber Mais

Libpd:

- Livro: Making Musical Apps - Peter Brinkmann
- Site: libpd.cc
- GitHub: github.com/libpd
- Código Fonte:

```
$ git clone https://github.com/libpd/libpd.git
```

Sistemas Musicais Multiagentes



Figura: Um Sistema Multiagentes.

Sistemas Musicais Multiagentes

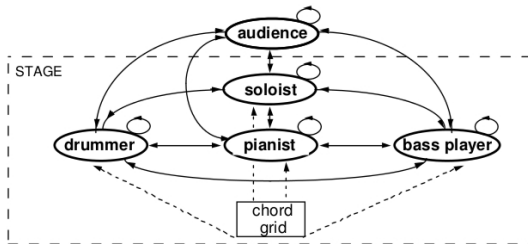


Figure 1 - Interaction model of a jazz ensemble

Generalização de um Sistema Musical Multiagentes [Thomaz e Queiroz, 2009]

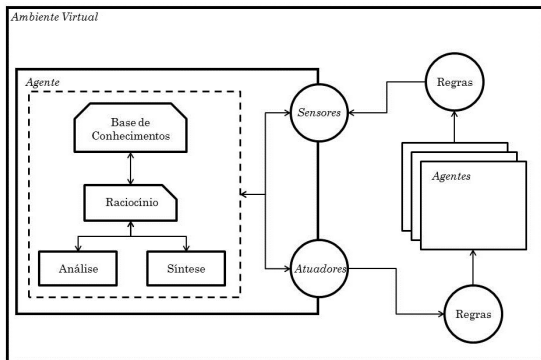


Figura: Agente Musical e Ambiente

Arcabouço Ensemble

O Arcabouço Ensemble fornece ferramentas multiplataforma para a implementação de Sistemas Musicais Multiagentes.

Código e Documentação

- GitHub: github.com/phrb/ensemble
- Google Code: code.google.com/p/musicalagents/

Arquitetura de Aplicação

Uma aplicação do Ensemble é composta de:

Arquitetura de Aplicação

Uma aplicação do Ensemble é composta de:

- Um Agente Ambiente e um Mundo;

Arquitetura de Aplicação

Uma aplicação do Ensemble é composta de:

- Um Agente Ambiente e um Mundo;
- Agentes e seus Raciocínios, Sensores e Atuadores;

Arquitetura de Aplicação

Uma aplicação do Ensemble é composta de:

- Um Agente Ambiente e um Mundo;
- Agentes e seus Raciocínios, Sensores e Atuadores;
- Servidores de Eventos;

Arquitetura de Aplicação

Uma aplicação do Ensemble é composta de:

- Um Agente Ambiente e um Mundo;
- Agentes e seus Raciocínios, Sensores e Atuadores;
- Servidores de Eventos;
- Entradas e Saídas do Ambiente Virtual;

Arquitetura de Aplicação

Uma aplicação do Ensemble é composta de:

- Um Agente Ambiente e um Mundo;
- Agentes e seus Raciocínios, Sensores e Atuadores;
- Servidores de Eventos;
- Entradas e Saídas do Ambiente Virtual;
- Interfaces;

Arquitetura de Aplicação

```
1 <ENSEMBLE NAME="Living Melodies">
2
3   <GLOBAL_PARAMETERS CLOCK_MODE="CLOCK_USER" PROCESS_MODE="BATCH" OSC="FALSE" />
4
5   <!-- ***** ENVIRONMENT AGENT ***** -->
6
7   <ENVIRONMENT_AGENT_CLASS NAME="ENVIRONMENT" CLASS="ensemble.apps.lm.LM_Environment">
8     <ARG NAME="WAIT_TIME_TURN" VALUE="300" />
9     <ARG NAME="WAIT_ALL_AGENTS" VALUE="TRUE" />
10    <WORLD CLASS="ensemble.apps.lm.LM_World">
11      <LAW CLASS="ensemble.movement.MovementLaw" />
12      <ARG NAME="dimensions" VALUE="2" />
13      <ARG NAME="structure" VALUE="continuous" />
14      <ARG NAME="form" VALUE="cube:200:loop" />
15    </WORLD>
16  </ENVIRONMENT_AGENT_CLASS>
17
18  <!-- ***** MUSICAL AGENT CLASSES ***** -->
19
20  <MUSICAL_AGENT_CLASS NAME="LM_AGENT" CLASS="ensemble.apps.lm.LM_MusicalAgent" />
21
22  <!-- ***** MUSICAL AGENT INSTANCES ***** -->
23
24  <MUSICAL_AGENT NAME="creature" CLASS="LM_AGENT" QUANTITY="30" />
25
26 </ENSEMBLE>
```

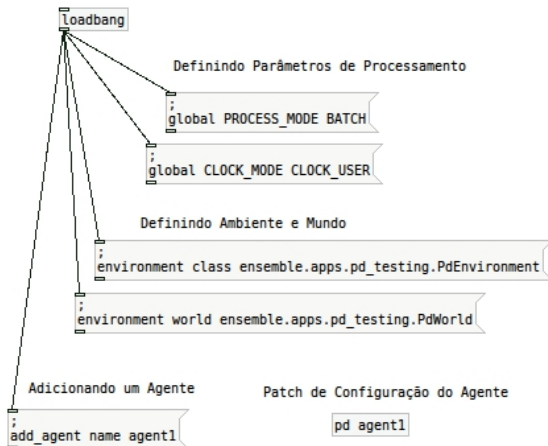
Figura: Arquivo de Configuração do Ensemble

Interface com o Pure Data

Objetivo:

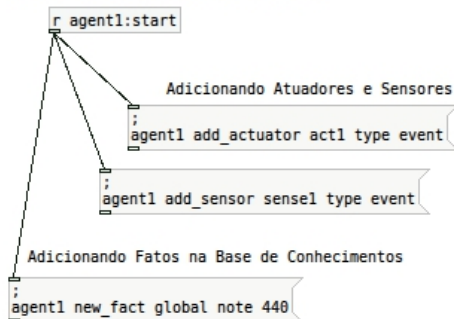
Mapear os conceitos e estruturas do Ensemble a objetos num patch, permitindo a criação e configuração de aplicações do Ensemble a partir do ambiente do Pure Data.

Arquitetura de Aplicação



Arquitetura de Aplicação

Símbolo Para Inicialização do Agente



The interface displays a hierarchical tree on the left and configuration fields on the right.

Tree Structure:

- Ensemble
 - Environment
 - World
 - event
 - agent1
 - KnowledgeBase
 - sense1
 - Reasoning
 - actself
 - senseself
 - act1

Configuration Fields:

- NAME:** Reasoning
- CLASS:** ensemble.apps.pd_testing.PdReasoning
- STATE:** INITIALIZED
- TYPE:** REASONING

Parameters Table:

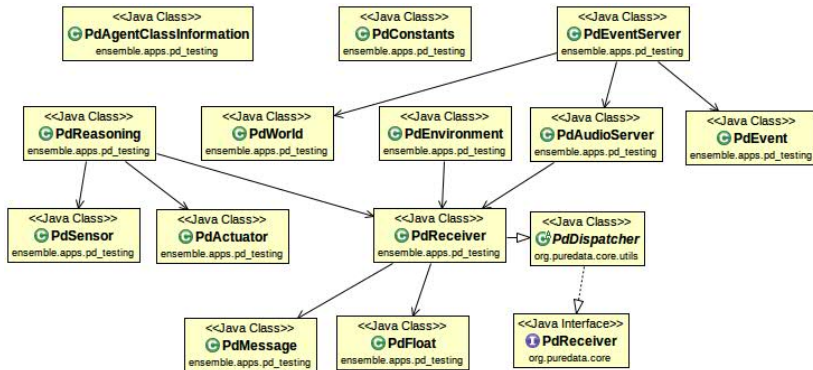
NAME	VALUE
subpatch_name	agent1

Buttons:

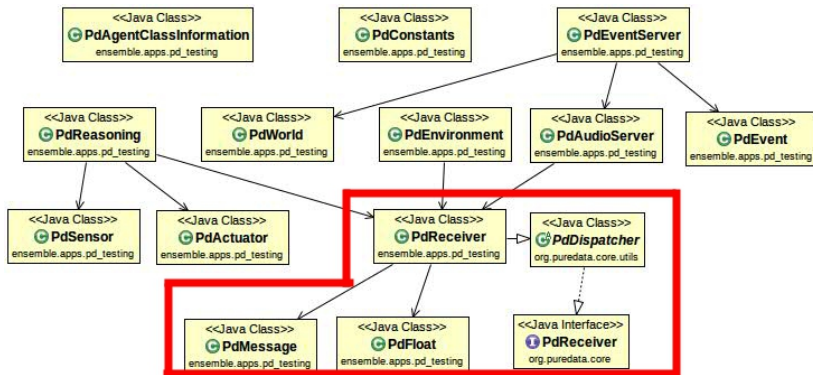
- Remove Component
- Send
- Start Simulation
- Stop Simulation

Command: [Empty text field]

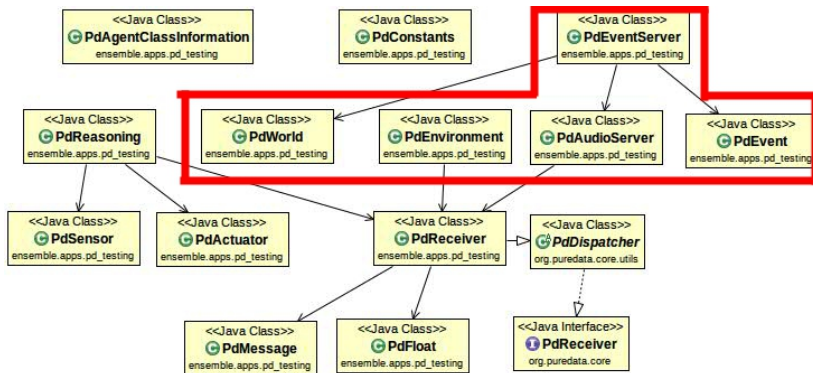
A Interface



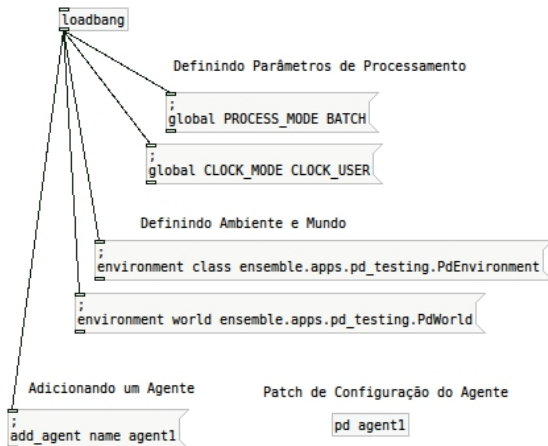
PdReceiver



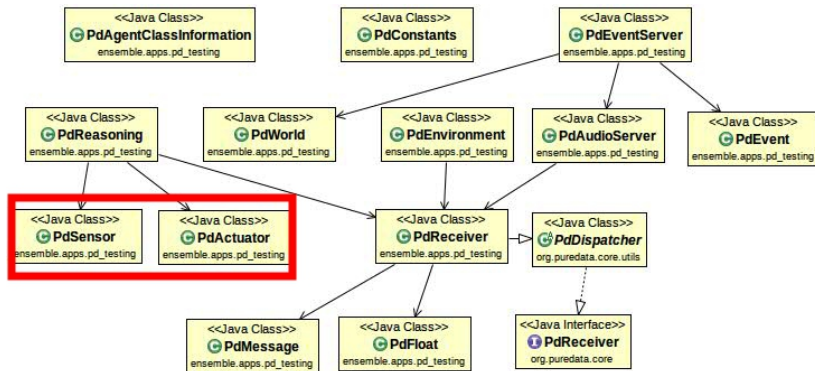
Servidores de Áudio e Eventos



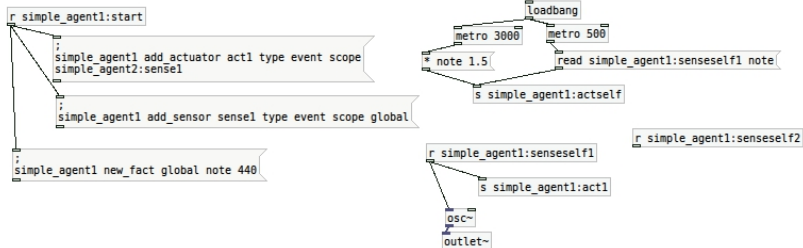
Servidores de Áudio e Eventos



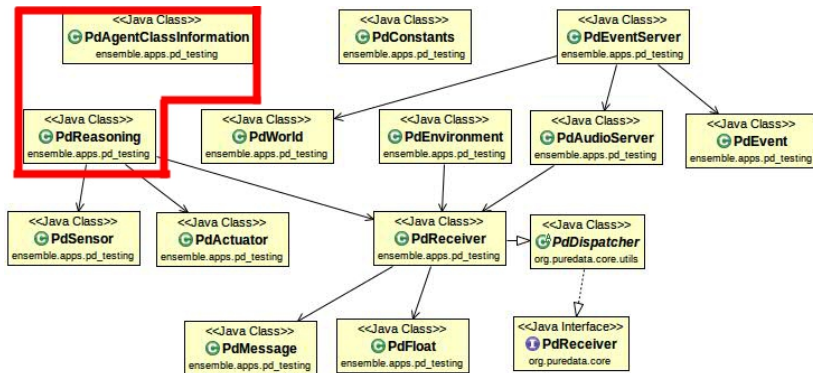
Sensores e Atuadores



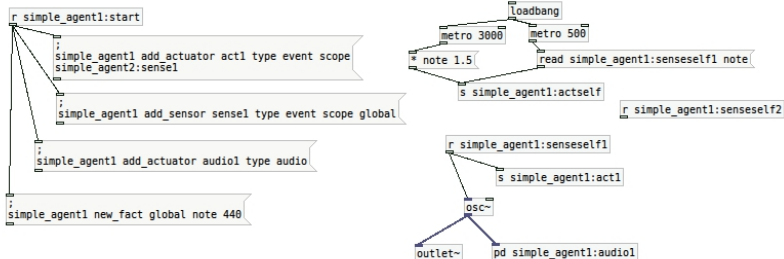
Sensores e Atuadores



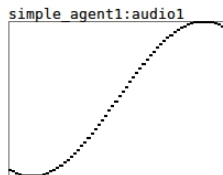
Agentes e Raciocínios



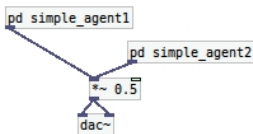
Entrada e Saída de Áudio



Entrada e Saída de Áudio



Entrada e Saída de Áudio





Brinkmann et al. (2011)

Embedding pure data with libpd

Proceedings of the Pure Data Convention



L. F. Thomaz e M. Queiroz (2009)

A framework for musical multiagent systems

*Proceedings of the SMC 2009 - 6th Sound and Music Computing
Conference 119–124*

Obrigado!