Andante: Composition and Performance with Mobile Musical Agents

Leo Kazuhiro Ueda
Fabio Kon

{lku,kon}@ime.usp.br
http://gsd.ime.usp.br/andante

Distributed Systems Research Group
Department of Computer Science
Institute of Mathematics and Statistics
University of São Paulo
Brazil

Project supported by CAPES and CNPq, Brazil

Outline

- Introduction
- Mobile Agents
- Mobile Musical Agents
- Andante infrastructure
- Applications
- Future work

Introduction

- Composers always looking for new forms of music composition and performance
- Mobile Agents → Mobile Musical Agents
- Goals:
  - Build an infrastructure for the development of musical applications using mobile agents (the Andante infrastructure)
  - Create such applications
  - Attract the interest of musicians and scientists

Mobile Agent

- Autonomous computer program capable of migrating from one machine to another
- Transfers its code and state
- Heterogeneous network
- May react to changes on the host environment
- Recently explored concept

Mobile Musical Agent

- Mobile agent that participates in the production of distributed music
- Agents get together in network nodes to make music
  - Analog to musicians getting together on a stage to make music

Mobile Musical Agent

- Each agent can perform one or more of the following actions:
  - Encapsulate an algorithm
  - Interact with other agents and/or real musicians
  - React to sensor
  - Migrate to another stage

Mobile Musical Agent

- Example
  - Distributed music
    - Room with several computers connected by a network
    - Each computer equipped with motion sensors and hosts a few agents
    - Agents communicate to each other and play a distributed music piece
    - A specific agent receives information from the sensor in order to follow someone who walks around the room

Mobile Musical Agent

- Example
  - Collaborative music
    - Each user implements an agent and send it to a network node
    - Agents in the same node interact
    - Everybody listens to the resulting sound

The Andante Infrastructure

- Open-source software infrastructure to build and run applications based on mobile musical agents
- Intended to run on different operating systems (such as Linux, MacOS, and Windows)
- Create a community where composers and programmers collaborate do develop the system
- Code available under the GPL

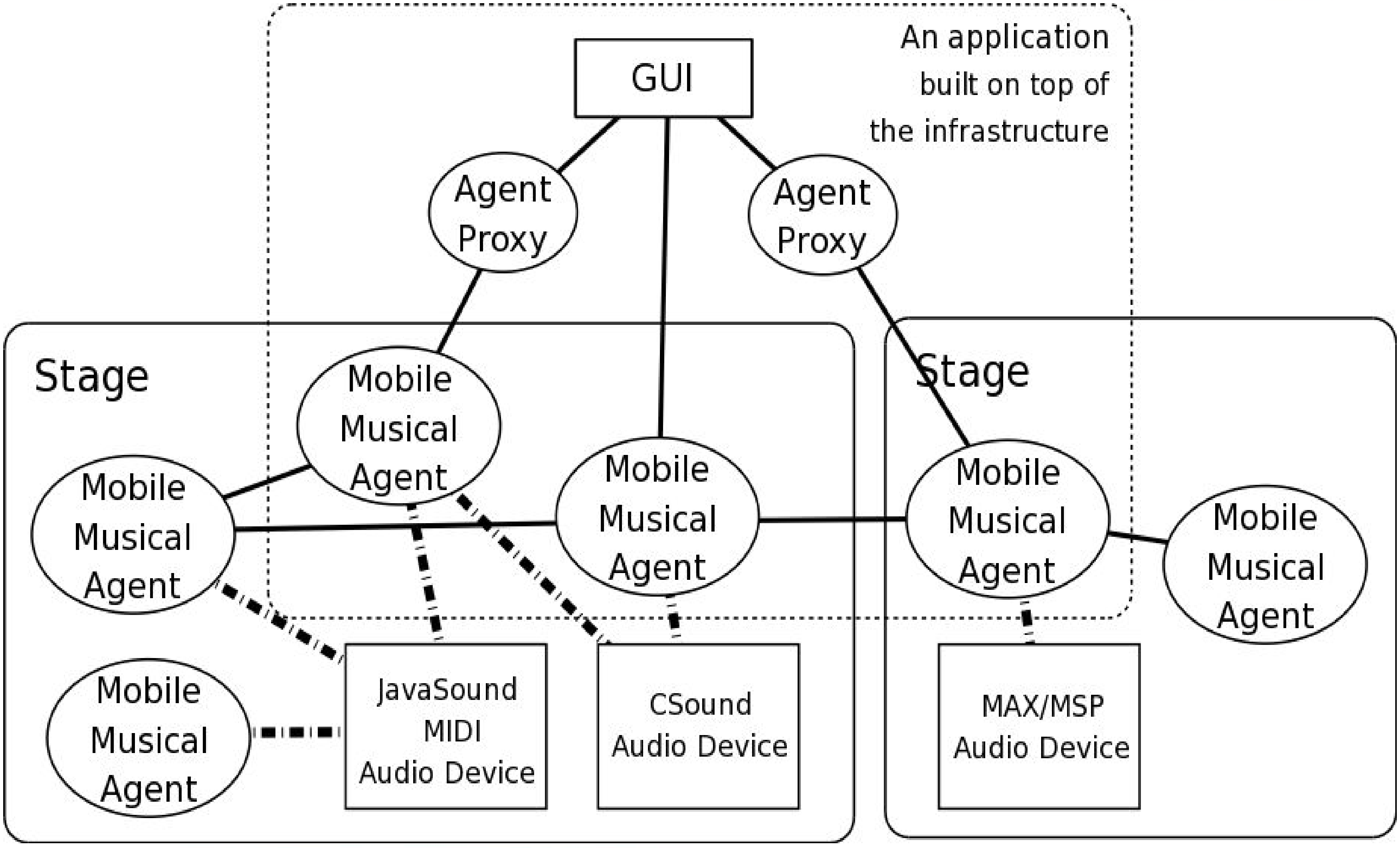
Used Technologies

- Java
  - Platform independence
  - Java Swing and Sound API
- CORBA
  - Allows the use of other programming languages
- Aglets
  - Mobile agent system support for Java
- MIDI (Java Sound API)
- MAX/MSP (only preliminary experiments)

Architecture

- *Stage* hosts agents
- *Agents* perform actions in a Stage
- *Audio Device* allows agents to produce sound
- *External program* provides human interaction with the agents
- *Agent Proxy* offers agent location transparency

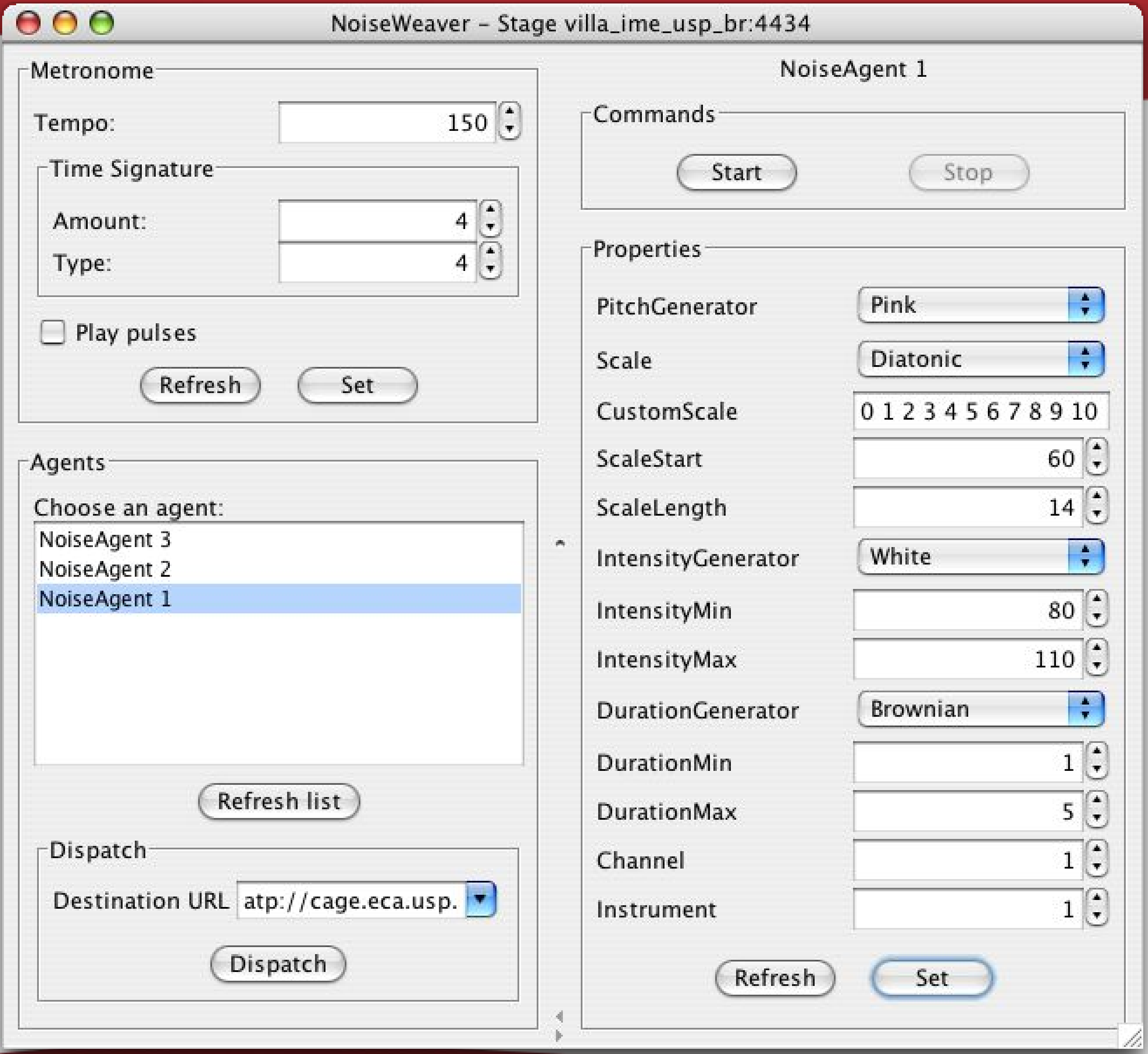
Architecture



Application: NoiseWeaver

- Generates and plays stochastic music in real-time
- One kind of agent that generates a stochastic melody
  - Pitch, intensity, and duration are determined by number generators based on $1/f^b$ noises
  - User-defined parameters determine the way the melody is generated
- Graphical interface controls parameters
- Metronome synchronizes agents

NoiseWeaver



NoiseWeaver – Stage villa_ime_usp_br:4434

Metronome

Tempo: 150

Time Signature

Amount: 4

Type: 4

□ Play pulses

Agents

Choose an agent:

NoiseAgent 3
NoiseAgent 2
NoiseAgent 1

Dispatch

Destination URL atp://cage.eca.usp.

NoiseAgent 1

Commands

Properties

PitchGenerator: Pink

Scale: Diatonic

CustomScale: 0 1 2 3 4 5 6 7 8 9 10

ScaleStart: 60

ScaleLength: 14

IntensityGenerator: White

IntensityMin: 80

IntensityMax: 110

DurationGenerator: Brownian

DurationMin: 1

DurationMax: 5

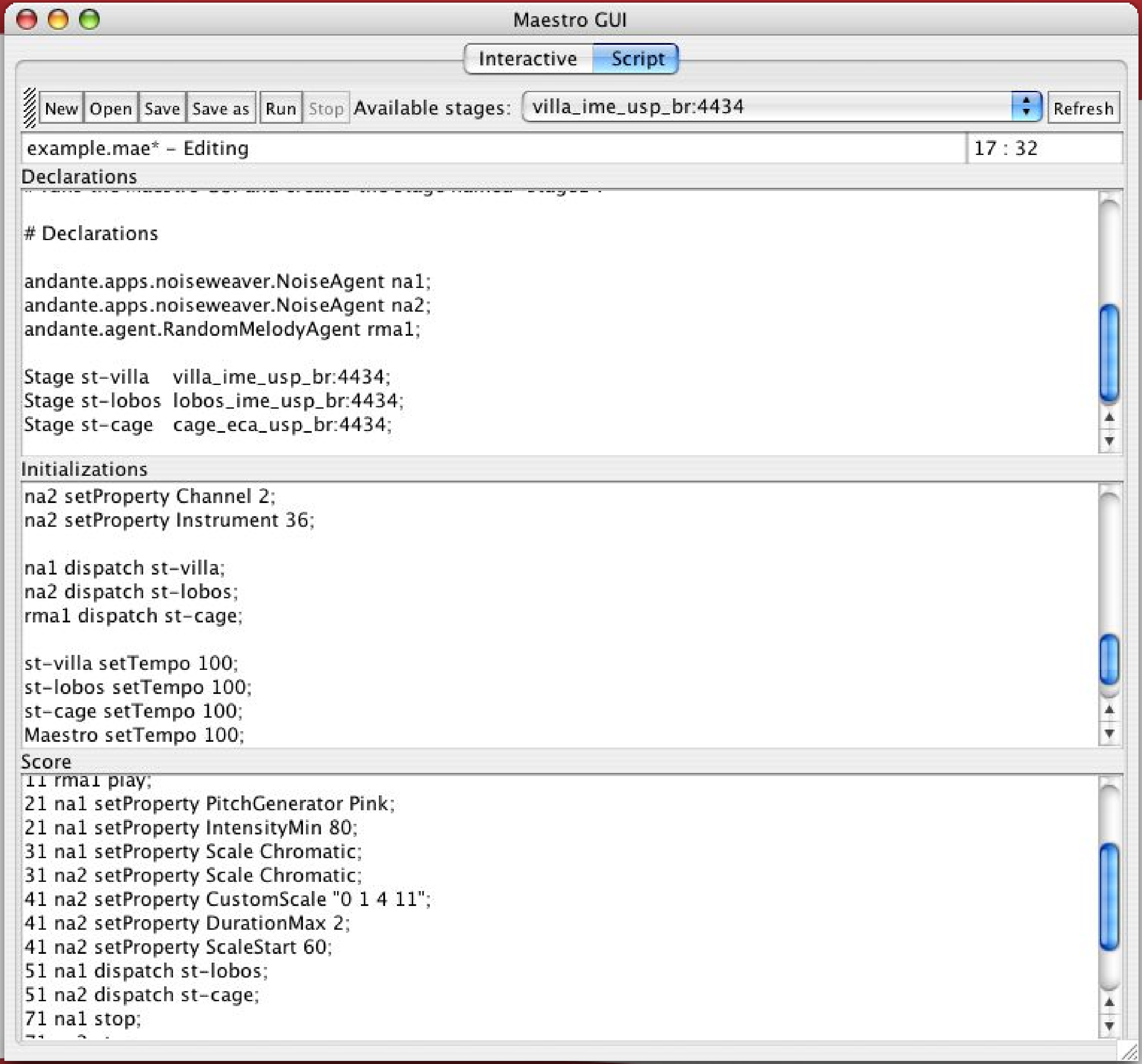
Channel: 1

Instrument: 1

Application: Maestro

- Allows a distributed collection of agents to be controlled by a script
- Script gives time-stamped commands to agents
- Can send commands to any kind of Andante agent (via Java Reflection API)
- Graphical interface to edit and run scripts
- Generic interface to control agents interactively (similar to NoiseWeaver)

Maestro Script GUI



example.mae* – Editing

17 : 32

Declarations

```
# Declarations

andante.apps.noiseweaver.NoiseAgent na1;
andante.apps.noiseweaver.NoiseAgent na2;
andante.agent.RandomMelodyAgent rma1;

Stage st-villa   villa_ime_usp_br:4434;
Stage st-lobos  lobos_ime_usp_br:4434;
Stage st-cage   cage_eca_usp_br:4434;
```

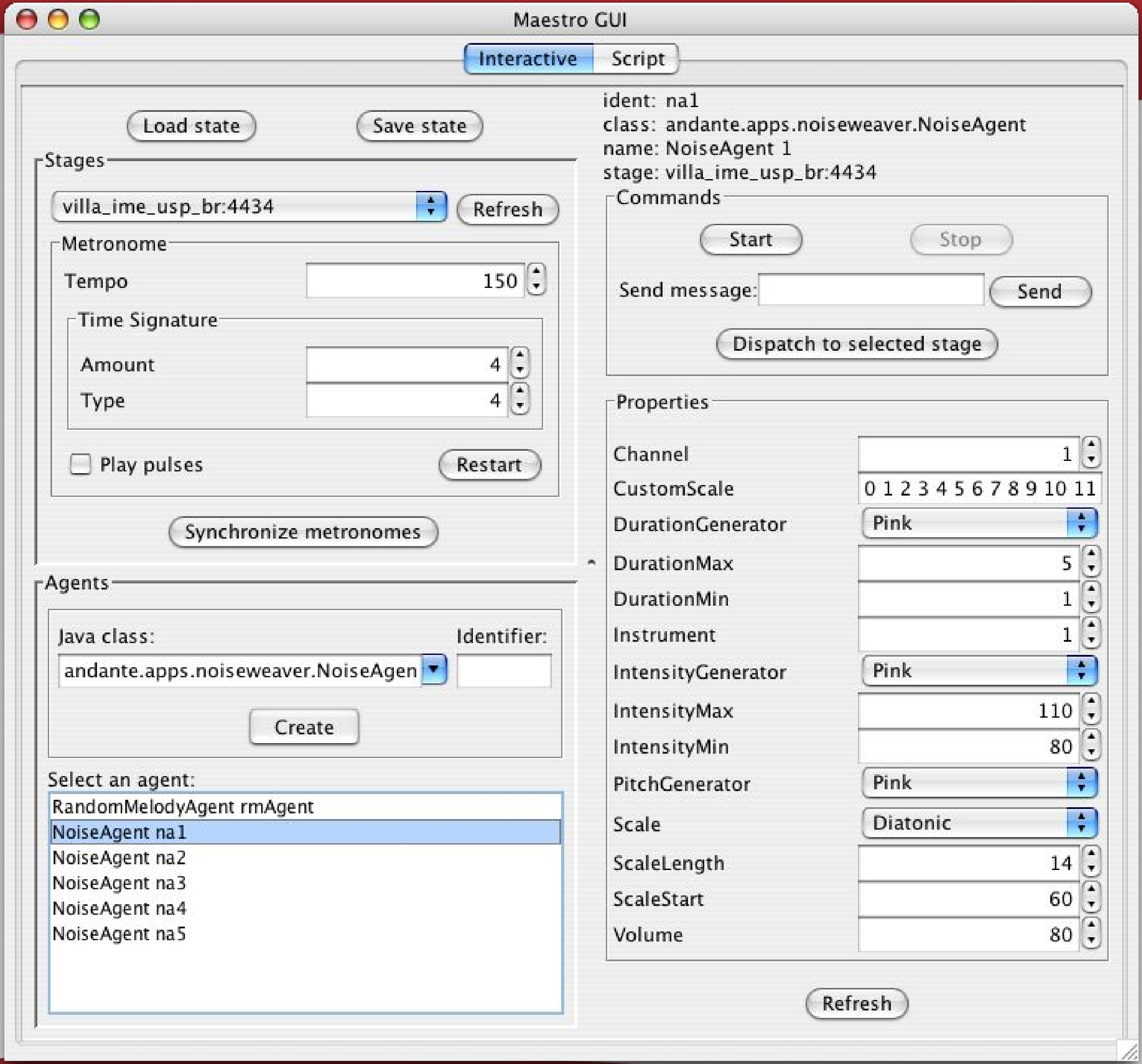
Initializations

```
na2 setProperty Channel 2;
na2 setProperty Instrument 36;

na1 dispatch st-villa;
na2 dispatch st-lobos;
rma1 dispatch st-cage;

st-villa setTempo 100;
st-lobos setTempo 100;
st-cage setTempo 100;
Maestro setTempo 100;
```

Score

```
11 rma1 play;
21 na1 setProperty PitchGenerator Pink;
21 na1 setProperty IntensityMin 80;
31 na1 setProperty Scale Chromatic;
31 na2 setProperty Scale Chromatic;
41 na2 setProperty CustomScale "0 1 4 11";
41 na2 setProperty DurationMax 2;
41 na2 setProperty ScaleStart 60;
51 na1 dispatch st-lobos;
51 na2 dispatch st-cage;
71 na1 stop;
```

Maestro Interactive GUI



ident: na1
class: andante.apps.noiseweaver.NoiseAgent
name: NoiseAgent 1
stage: villa_ime_usp_br:4434

Stages: villa_ime_usp_br:4434

Metronome

Tempo: 150

Time Signature

Amount: 4
Type: 4

□ Play pulses

Agents

Java class: andante.apps.noiseweaver.NoiseAgen
Identifier:

Select an agent:

- RandomMelodyAgent rmAgent
- NoiseAgent na1
- NoiseAgent na2
- NoiseAgent na3
- NoiseAgent na4
- NoiseAgent na5

Commands

Send message:

Properties

Channel	1
CustomScale	0 1 2 3 4 5 6 7 8 9 10 11
DurationGenerator	Pink
DurationMax	5
DurationMin	1
Instrument	1
IntensityGenerator	Pink
IntensityMax	110
IntensityMin	80
PitchGenerator	Pink
Scale	Diatonic
ScaleLength	14
ScaleStart	60
Volume	80

Maestro script

```
# Declarations

NoiseAgent a1;
NoiseAgent a2;
Stage s1 villa:4434;
Stage s2 lobos:4434;
--
# Initializations

a1 setProperty Scale Diatonic;
a1 setProperty ScaleStart 36;
a1 setProperty Channel 1;
a1 setProperty Instrument 1;
a2 setProperty PitchGenerator Pink;
a2 setProperty Scale Diatonic;
a2 setProperty ScaleStart 72;
a2 setProperty Channel 2;
a2 setProperty Instrument 36;
a1 dispatch s1;
a2 dispatch s2;
s1 setTempo 150;
s2 setTempo 150;
--
```

```
# Score

1  a1 play;
5  a2 play;
13 a1 setProperty IntensityMin 80;
13 a1 setProperty Scale Chromatic;
17 a2 setProperty DurationMax 2;
17 a2 setProperty ScaleStart 60;
25 a1 dispatch s2;
25 a2 dispatch s1;
37 a1 stop;
37 a2 stop;
--
```

Future Work

- Quality of Service
- Work on other sound technologies: MAX/MSP, jMax, CSound, Siren, jMusic
- New applications that explore the mobility of the agents
- Move focus to music creation

For more information

- Andante Web site:

  `http://gsd.ime.usp.br/andante`

- Contact: Leo Ueda <`lku@ime.usp.br`>, Fabio Kon <`kon@ime.usp.br`>

- Join us: the Andante project is seeking the help of musicians and software developers. There is a lot to be done!