Medusa - A Distributed Sound Environment

Flávio Luiz Schiavoni, Marcelo Queiroz, Fernando Iazzetta

USP - University of São Paulo - Brazil
Mobile Interactive Musical Processes
http://www.eca.usp.br/mobile/portal/

May 07, 2011

Outline

- Scenario

Outline

- Scenario
- Goals

Outline

- Scenario
- Goals
- Related works

Outline

- Scenario
- Goals
- Related works
- Methodology

Outline

- Scenario
- Goals
- Related works
- Methodology
- Desirable Features

Outline

- Scenario
- Goals
- Related works
- Methodology
- Desirable Features
- System Architecture

Outline

- Scenario
- Goals
- Related works
- Methodology
- Desirable Features
- System Architecture
- Results

Outline

- Scenario
- Goals
- Related works
- Methodology
- Desirable Features
- System Architecture
- Results
- Future works

Scenario

This project is part of Mobile (Interactive Musical Processes) research group.
The research group involves Musicians, Electrical Engineers, Computer Scientists, Visual Artists, ...

Goals

- Speed up network music setup
- Create a network music environment
- Rich range of interaction possibilities
- Local Area Network as Case Study

Related work

Some related work address the problem of synchronous music communication between networked computers, such as

- OSC [Lazzaro and Wawrzynek2001]
- NetJack [Carôt et al.2009]
- SoundJack [Carôt et al.2006]
- JackTrip [Cáceres and Chafe2009b, Cáceres and Chafe2009a]
- eJamming [Renaud et al.2007]
- Otherside [Anagnostopoulos2009]
- LDAS [Sæbø and Svensson2006]
- ReWire [Kit2010].

Methodology

Our Methodology intend to join different research areas to design a sound environment:

- Distributed Systems

Methodology

Our Methodology intend to join different research areas to design a sound environment:

- Distributed Systems
- Computer Network

Methodology

Our Methodology intend to join different research areas to design a sound environment:

- Distributed Systems
- Computer Network
- Musical Computing

Methodology

Our Methodology intend to join different research areas to design a sound environment:

- Distributed Systems
- Computer Network
- Musical Computing
- Software engineering

Methodology

Our Methodology intend to join different research areas to design a sound environment:

- Distributed Systems
- Computer Network
- Musical Computing
- Software engineering
- Network Music Performance

Methodology

- Map desirable features

Methodology

- Map desirable features
- Verify priorities and dependence

Methodology

- Map desirable features
- Verify priorities and dependence
- Architectural view of features

Methodology

- Map desirable features
- Verify priorities and dependence
- Architectural view of features
- Implementation / Prototype

Methodology

- Map desirable features
- Verify priorities and dependence
- Architectural view of features
- Implementation / Prototype
- Validation

Desirable Features...

- Transparency

Desirable Features...

- Transparency
- Heterogeneity

Desirable Features...

- Transparency
- Heterogeneity
- Graphical display of status and messages

Desirable Features...

- Transparency
- Heterogeneity
- Graphical display of status and messages
  - Latency and communication status

Desirable Features...

- Transparency
- Heterogeneity
- Graphical display of status and messages
  - Latency and communication status
  - Network status

Desirable Features...

- Transparency
- Heterogeneity
- Graphical display of status and messages
  - Latency and communication status
  - Network status
  - Input/Output status

Desirable Features...

- Transparency
- Heterogeneity
- Graphical display of status and messages
  - Latency and communication status
  - Network status
  - Input/Output status
  - IO stream amplitudes

Desirable Features...

- Transparency
- Heterogeneity
- Graphical display of status and messages
  - Latency and communication status
  - Network status
  - Input/Output status
  - IO stream amplitudes
  - CPU Meter

Desirable Features...

- Transparency
- Heterogeneity
- Graphical display of status and messages
  - Latency and communication status
  - Network status
  - Input/Output status
  - IO stream amplitudes
  - CPU Meter
  - Memory Meter

Desirable Features...

- Transparency
- Heterogeneity
- Graphical display of status and messages
  - Latency and communication status
  - Network status
  - Input/Output status
  - IO stream amplitudes
  - CPU Meter
  - Memory Meter
- Multiple IO information types

Desirable Features...

- Transparency
- Heterogeneity
- Graphical display of status and messages
  - Latency and communication status
  - Network status
  - Input/Output status
  - IO stream amplitudes
  - CPU Meter
  - Memory Meter
- Multiple IO information types
  - Audio

Desirable Features...

- Transparency
- Heterogeneity
- Graphical display of status and messages
  - Latency and communication status
  - Network status
  - Input/Output status
  - IO stream amplitudes
  - CPU Meter
  - Memory Meter
- Multiple IO information types
  - Audio
  - MIDI

Desirable Features...

- Transparency
- Heterogeneity
- Graphical display of status and messages
  - Latency and communication status
  - Network status
  - Input/Output status
  - IO stream amplitudes
  - CPU Meter
  - Memory Meter
- Multiple IO information types
  - Audio
  - MIDI
  - Control Messages

Desirable Features...

- Transparency
- Heterogeneity
- Graphical display of status and messages
  - Latency and communication status
  - Network status
  - Input/Output status
  - IO stream amplitudes
  - CPU Meter
  - Memory Meter
- Multiple IO information types
  - Audio
  - MIDI
  - Control Messages
  - User text messages

Desirable Features

- Legacy software integration [Young2001]

Desirable Features

- Legacy software integration [Young2001]
  - Audio integration

Desirable Features

- Legacy software integration [Young2001]
  - Audio integration
  - MIDI integration

Desirable Features

- Legacy software integration [Young2001]
  - Audio integration
  - MIDI integration
  - Control integration

Desirable Features

- Legacy software integration [Young2001]
  - Audio integration
  - MIDI integration
  - Control integration
- Sound processing capabilities [Chafe et al.2000]

Desirable Features

- Legacy software integration [Young2001]
  - Audio integration
  - MIDI integration
  - Control integration
- Sound processing capabilities [Chafe et al.2000]
  - Master Mixer [Cáceres and Chafe2009a]

Desirable Features

- Legacy software integration [Young2001]
  - Audio integration
  - MIDI integration
  - Control integration
- Sound processing capabilities [Chafe et al.2000]
  - Master Mixer [Cáceres and Chafe2009a]
  - Silence Detection [Bolot and García1996]

Desirable Features

- Legacy software integration [Young2001]
  - Audio integration
  - MIDI integration
  - Control integration
- Sound processing capabilities [Chafe et al.2000]
  - Master Mixer [Cáceres and Chafe2009a]
  - Silence Detection [Bolot and García1996]
  - Data compression [Chafe et al.2000]

Desirable Features

- Legacy software integration [Young2001]
  - Audio integration
  - MIDI integration
  - Control integration
- Sound processing capabilities [Chafe et al.2000]
  - Master Mixer [Cáceres and Chafe2009a]
  - Silence Detection [Bolot and García1996]
  - Data compression [Chafe et al.2000]
  - Loopback [Cáceres and Chafe2009a]

Development API

- C++

Development API

- C++
- Jack

Development API

- C++
- Jack
- QT

Development API

- C++
- Jack
- QT
- SCTP

System Architecture

- Peer Connection: No Central Server (Sources and Sinks)

System Architecture

- Peer Connection: No Central Server (Sources and Sinks)
- Layered architecture in each computer (Node)

System Architecture

- Peer Connection: No Central Server (Sources and Sinks)
- Layered architecture in each computer (Node)
- Network messages to ensure environment integrity

Node Architecture

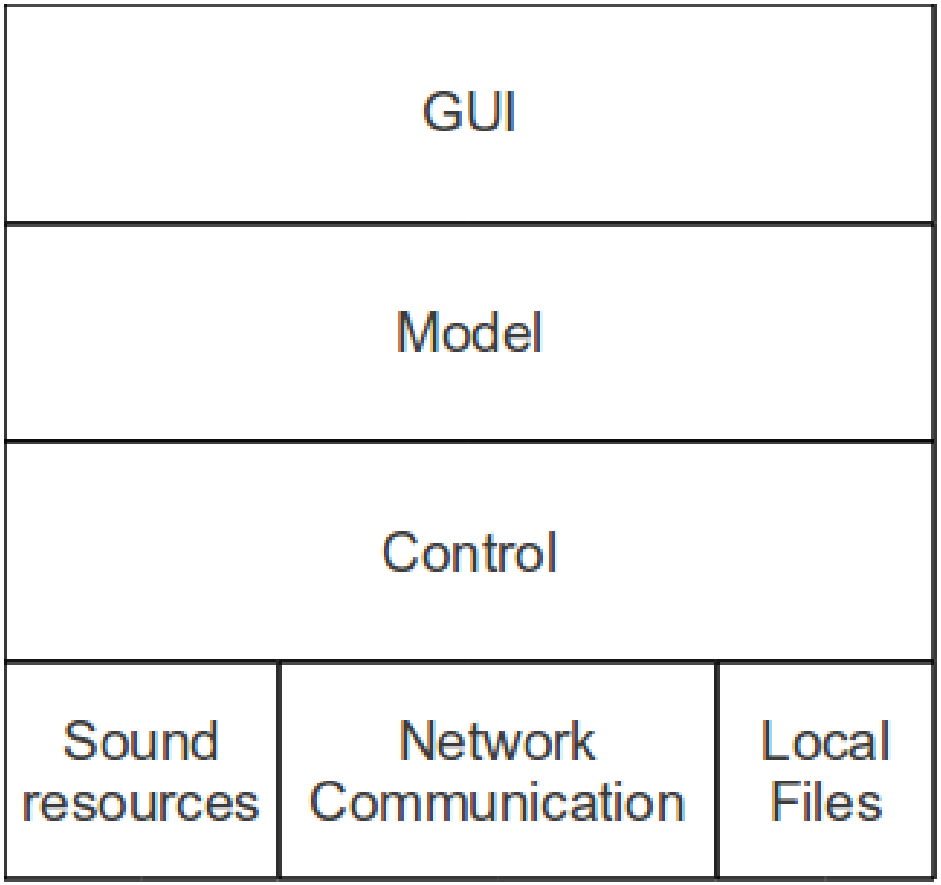


Figure: Node Architecture

Node Architecture - Resources

- Network Communication
  - UDP for BroadCast Communication
  - TCP for Unicast Communication
  - SCTP for Streaming

Node Architecture - Resources

- Network Communication
  - UDP for BroadCast Communication
  - TCP for Unicast Communication
  - SCTP for Streaming
- Sound Resources
  - Jack Connection API
  - JackInput (Singleton)
  - JackOutput (Collection)

Node Architecture - Resources

- Network Communication
  - UDP for BroadCast Communication
  - TCP for Unicast Communication
  - SCTP for Streaming
- Sound Resources
  - Jack Connection API
  - JackInput (Singleton)
  - JackOutput (Collection)
- Configuration File

Node Architecture - Control

- Main Control (Facade / Bridge)

Node Architecture - Control

- Main Control (Facade / Bridge)
- Sound Control (Sound Resources / Jack API)

Node Architecture - Control

- Main Control (Facade / Bridge)
- Sound Control (Sound Resources / Jack API)
- Network Control (Network Communication)

Node Architecture - Control

- Main Control (Facade / Bridge)
- Sound Control (Sound Resources / Jack API)
- Network Control (Network Communication)
- Log Control (Log file)

Node Architecture - Control

- Main Control (Facade / Bridge)
- Sound Control (Sound Resources / Jack API)
- Network Control (Network Communication)
- Log Control (Log file)
- Message Control (Environment Messages)

Node Architecture - Control

- Main Control (Facade / Bridge)
- Sound Control (Sound Resources / Jack API)
- Network Control (Network Communication)
- Log Control (Log file)
- Message Control (Environment Messages)
- Configuration Control (Configuration File)

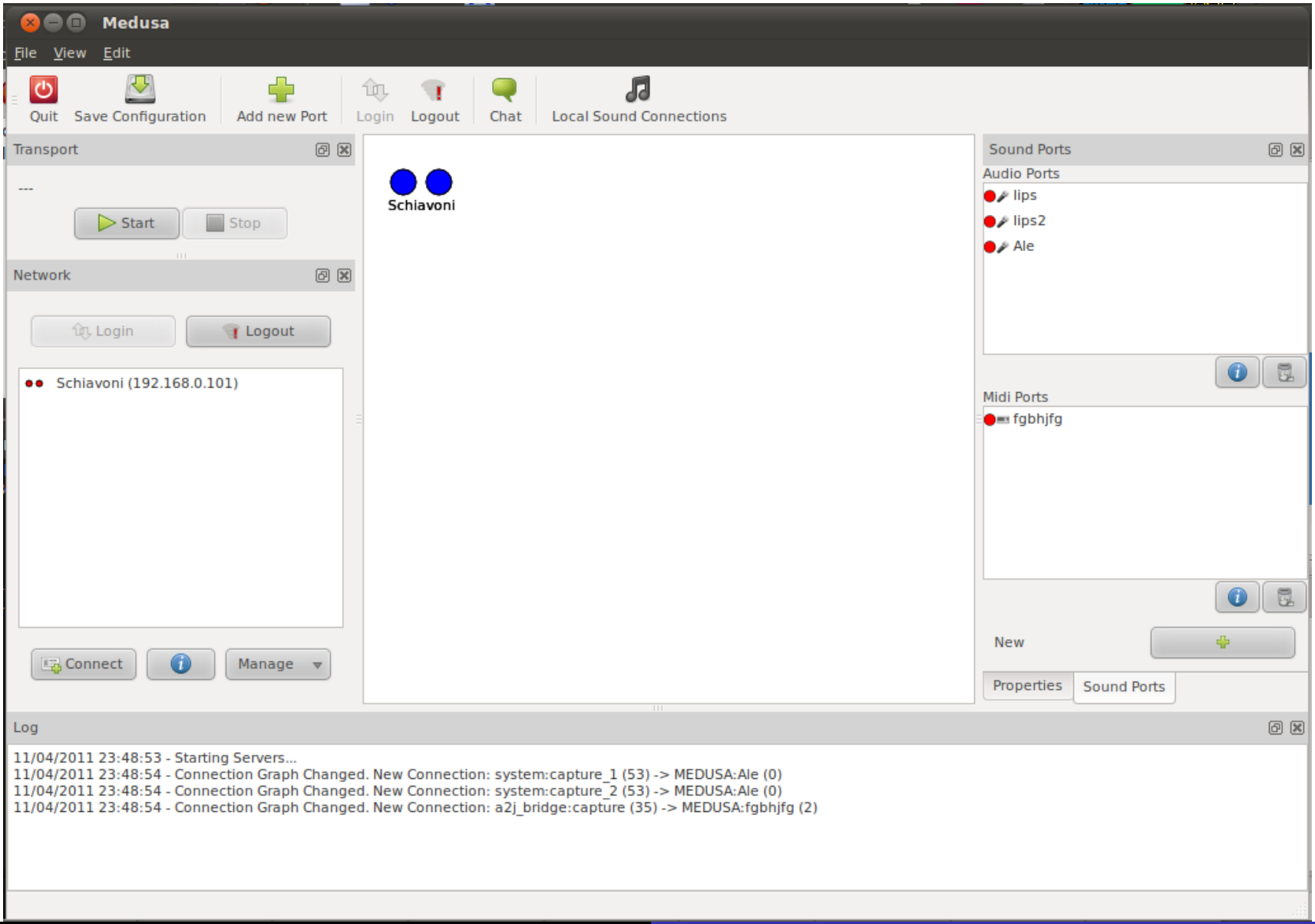
Node Architecture - Model

Local Settings = Sound Settings + Network Settings
Sound Settings = SoundPorts + SoundConnections
Environment = All Nodes + Global Node Connections

Node Architecture - Model

Local Settings = Sound Settings + Network Settings
Sound Settings = SoundPorts + SoundConnections
Environment = All Nodes + Global Node Connections
This model allows heterogeneous nodes and is easily extendable.

Node Architecture - GUI



Environment Maintenance

- Action Messages (Unicast)
  - add a port
  - connect a node

Environment Maintenance

- Action Messages (Unicast)
  - add a port
  - connect a node
- Information Messages (BroadCast)
  - port added
  - node connected

Environment Maintenance - Messages

HI_GUYS — HI_THERE — BYE

Environment Maintenance - Messages

HI_GUYS — HI_THERE — BYE
START_TRANSPORT — STOP_TRANSPORT

Environment Maintenance - Messages

HI_GUYS — HI_THERE — BYE
START_TRANSPORT — STOP_TRANSPORT
CONNECT_NODE — NODE_CONNECTED —
DISCONNECT_NODE — NODE_DISCONNECTED

Environment Maintenance - Messages

HI_GUYS — HI_THERE — BYE
START_TRANSPORT — STOP_TRANSPORT
CONNECT_NODE — NODE_CONNECTED —
DISCONNECT_NODE — NODE_DISCONNECTED
ADD_PORT — PORT_ADDED — REMOVE_PORT —
PORT_REMOVED

Environment Maintenance - Messages

HI_GUYS — HI_THERE — BYE
START_TRANSPORT — STOP_TRANSPORT
CONNECT_NODE — NODE_CONNECTED — DISCONNECT_NODE — NODE_DISCONNECTED
ADD_PORT — PORT_ADDED — REMOVE_PORT — PORT_REMOVED
CONNECT_PORT — PORT_CONNECTED — DISCONNECT_PORT — PORT_DISCONNECTED

Environment Maintenance - Messages

HI_GUYS — HI_THERE — BYE
START_TRANSPORT — STOP_TRANSPORT
CONNECT_NODE — NODE_CONNECTED — DISCONNECT_NODE — NODE_DISCONNECTED
ADD_PORT — PORT_ADDED — REMOVE_PORT — PORT_REMOVED
CONNECT_PORT — PORT_CONNECTED — DISCONNECT_PORT — PORT_DISCONNECTED
CHAT

Environment Maintenance - Messages

HI_GUYS — HI_THERE — BYE
START_TRANSPORT — STOP_TRANSPORT
CONNECT_NODE — NODE_CONNECTED — DISCONNECT_NODE — NODE_DISCONNECTED
ADD_PORT — PORT_ADDED — REMOVE_PORT — PORT_REMOVED
CONNECT_PORT — PORT_CONNECTED — DISCONNECT_PORT — PORT_DISCONNECTED
CHAT
LOOP_BACK

Environment Maintenance - The Messages

```
<msg bufferSize="512" ip="192.168.0.101" msgType="0"
      name="Flavio" sampleRate="44100" port="40000">
 <outputs>
  <audioOutput name="My Output 1"/>
  <audioOutput name="My Output 2"/>
 </ouputs>
</msg>
```

Figure: A HI_GUYS Message

Results

- Messages may help Network Music configuration

Results

- Messages may help Network Music configuration
- Possibilities of heterogeneous node configuration

Results

- Messages may help Network Music configuration
- Possibilities of heterogeneous node configuration
- GUI and configuration file speed up configuration

Results

- Messages may help Network Music configuration
- Possibilities of heterogeneous node configuration
- GUI and configuration file speed up configuration
- Environment view gives instantaneous feedback

Results

- Messages may help Network Music configuration
- Possibilities of heterogeneous node configuration
- GUI and configuration file speed up configuration
- Environment view gives instantaneous feedback
- Difficulties in testing different network conditions

Future work

- Implementation of full desirable features list

Future work

- Implementation of full desirable features list
- Better thread-safe implementation

Future work

- Implementation of full desirable features list
- Better thread-safe implementation
- Testing other network protocols

Future work

- Implementation of full desirable features list
- Better thread-safe implementation
- Testing other network protocols
- Integration with NetJack / JackTrip

Acknowledgements

The authors would like to thank the support of the funding agencies CNPq and FAPESP - São Paulo Research Foundation (grant 2008/08623-8).

Thanks!

http://sourceforge.net/projects/medusa-audionet/
fls@ime.usp.br
Questions?

Thanks!


Ilias Anagnostopoulos.
2009.
The otherside web-based collaborative multimedia system.
In LAC, editor, *Proceedings of Linux Audio Conference 2009*, pages 131–137.


Jean-Chrysostome Bolot and Andrés Vega García.
1996.
Control mechanisms for packet audio in the internet.
In *INFOCOM '96. Fifteenth Annual Joint Conference of the IEEE Computer Societies. Networking the Next Generation. Proceedings IEEE*, pages 232 – 239 vol.1.


A. Carôt, U. Kramer, and G. Schuller.
2006.
Network music performance (NMP) in narrow band networks.
In *Proceedings of the 120th AES Convention*, Paris, France.


A. Carôt, T. Hohn, and C. Werner.
2009.

Netjack–remote music collaboration with electronic sequencers on the internet.
In *In Proceedings of the Linux Audio Conference*, page 118, Parma, Italy.

Chris Chafe, Scott Wilson, Al Leistikow, Dave Chisholm, and Gary Scavone.
2000.
A simplified approach to high quality music and sound over IP.
In *In Proceedings of the COST G-6 Conference on Digital Audio Effects (DAFX-00*, pages 159–164.

Juan-Pablo Cáceres and Chris Chafe.
2009a.
Jacktrip: Under the hood of an engine for network audio.
In *Proceedings of International Computer Music Conference*, page 509–512, San Francisco, California: International Computer Music Association.

Juan-Pablo Cáceres and Chris Chafe.

2009b.
Jacktrip/Soundwire meets server farm.
In *In Proceedings of the SMC 2009 - 6th Sound and Music Computing Conference*, pages 95–98, Porto, Portugal.


ReWire Software Development Kit.
2010.
Propellerhead software.
Stockholm, Sweden.


John Lazzaro and John Wawrzynek.
2001.
A case for network musical performance.
In *In Proceedings of the 11th international*, pages 157–166.
ACM Press.


Alain B. Renaud, Alexander Carôt, and Pedro Rebelo.
2007.
Networked music performance : State of the art.
In *Proceedings AES 30th International Conference*, Saariselkä, Finland.

Asbjørn Sæbø and U. Peter Svensson.
2006.
A low-latency full-duplex audio over IP streamer.
In *Proceedings of the Linux Audio Conference*, pages 25–31, Karlsruhe, Germany.

John P. Young.
2001.
Using the Web for live interactive music.
In *Proc. International Computer Music Conference*, pages 302–305, Habana, Cuba.