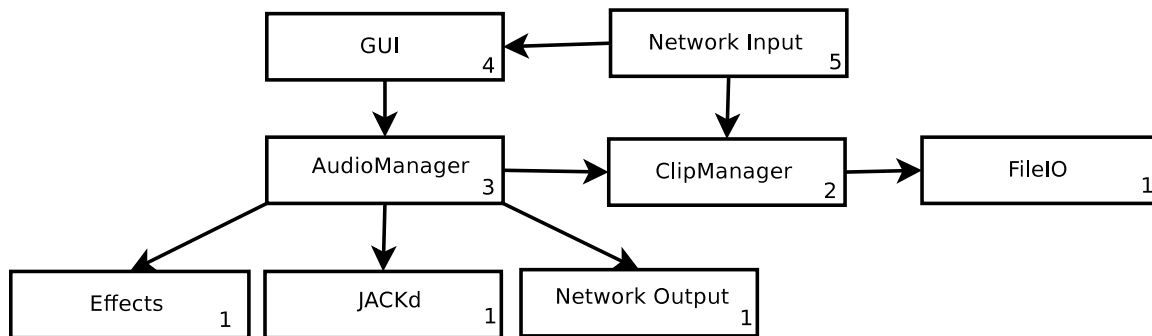


LoopBack Toplevel Design

Colin Stebbins Gordon (colin)

February 22, 2007

1 Components



- Network Input - This listens for requests from other instances of LoopBack, and accepts input from valid connections. When an incoming request to connect is recieved, it calls into the GUI for a response from the user. Once connections are created, it listens for new track info from other clients. If a track is requested to be added which the local machine doesn't have, it will call into the clip manager, and get a copy of the clip for the local machine from the remote machine.
- GUI - User interface. See UI mockup in Specifications Document for design. Responsible for allowing user to manage tracks. When a call is made into the GUI by the network input module, the GUI treats this input as if the user interacting directly with the UI performed the corresponding actions himself (e.g. adding a track, switching a track to looping).
- Clip Manager - Keeps track of what clips the local machine has. Manages creation of new clips, retrieval of clips, etc.
- FileIO - Responsible for getting clips to and from disk.
- Audio Manager - Main module, controlled by the GUI. Recieves input from the UI. It calls into the clip manager for getting (or creating) clips, propagates edit commands to connected clients via the network output module, calls into the effects module to enact effects, and spools audio to JACK.

- Effects - Apply effects - looping, pitch change, etc. Should be straightforward, as there are only a couple algorithms, which should be fairly well-documented online.
- Network Output - Propagates copies of user actions out across the network.
- jackd - Recieves and mixes multiple audio streams (external dependency).

2 External Dependencies

2.1 JACK

JACK will be responsible for mixing the audio output. If it doesn't support enough input streams, or if we can't spawn enough of our own threads for all the tracks, we'll have to mix the audio streams together ourselves, which will increase the complexity of the project.

2.2 Linux network support

This basically depends on Linux support for TCP being correct.

3 Task Breakdown and Group Org

GUI - Nate, Owen

Responsible for implementing the GUI. Need to create a smooth UI, and export hooks for the network input module to call into the GUI as if it were another local user.

Network Input and Output - Colin

Propagate and recieve user events to and from other instances of LoopBack over the network. Transfer and recieve clips not present on one or more machines. Requires designing a network protocol.

Audio Manager and JACK interface - Sean, Lincoln

Responsible for managing the internal structures representing tracks. Probably the largest module, and the most central. Must decide how to manage spooling audio from multiple tracks.

Effects - Brendan, Colin

Implementing speed/pitch distortion, per-track volume adjustment, looping setup.

Clip Management and FileIO - Josh, Kevin

Fast search and retrieval of clips. Choice of a coherent naming scheme. File codecs. Choice of internal representation of sounds (MIDI style, or sample based).

Architect - Kevin, Lincoln

Responsible for creating cohesive overall design, and ensuring that all implementers understand and

abide by it. Also responsible for getting feedback from implementers which may alter the design.

Documentation - Dominic

Responsible for documenting UI and network semantics.

PM - Dominic

Responsible for ensuring that the project is on schedule. Should set up meetings between project subgroups, and regular progress meetings.

Build System & Tools - Lincoln

Responsible for managing the SVN repository, makefiles, etc.

Testing - Tara, Brendan

Create testing plan for whole project, and write unit tests for each module. Work with tools coordinator to create one or more test targets, to make regression testing easy for implementers as they progress.

4 Schedule

- 2/26 - Pick project
- 3/5 - Updated top level design, corrected roles, following group input.
- 3/12 - Defined interfaces between modules.
- 3/19 - Completed designs for each module's internals; begin implementation of tier one components, and code for generating the GUIs appearance.
- 3/26 - Lower tier components completed. Begin clip manager, and portions of audio manager independent of clip manager. Begin creation of tests for first tier.
- 4/9 - GUI connected to audio manager.
- 4/16 - Network input complete; final testing begins.
- 4/23 - Continue testing & debugging.
- 4/30 - Continue testing & debugging.
- 5/17 - Demo

5 Assumptions

I omitted most of the optional features, except for adding effects in pseudo-realtime (a few seconds of processing before playback begins is reasonable).