1 Overview

The system is divided into two parts, the *client* and the *server*. The server runs on a single computer, connected to the various devices and listens to incoming Open Sound Control (OSC) messages and translates them to the appropriate control for the target device.

The client runs on any number of computers networked to the server and serves to translate MIDI into OSC which it then relays to the server. This is optional, users can interface with the server directly through OSC (via UDP) if desired.

1.1 Server

The server determines what instruments are available by reading files in the `Instruments` directory of the project folder; it assumes every file present represents an instrument currently connected and sets itself up to communicate to them based on the information in the file.

1.2 Client

Client machines communicate to the server using OSC over UDP. A translation layer is provided which accepts MIDI messages and turns them into OSC which it then sends to the server. Some form of inter-application MIDI routing is required for this to succeed, but allows the use of standard musical software to interface with the server.

When the client translation layer starts up it notifies the server of its presence on the network. The server then enumerates the instruments it is currently connected to and any methods they offer outside the standard interface. This step means the client layer is agnostic to changes on the server – instruments can come and go and controls can change without any updates needing to happen on the client end.

Currently the client automatically allocates MIDI channels, starting from 1, in the order they are presented by the server. This order is not necessariy guaranteed, but at the end of initialisation the client will print a list of all the instruments, their channel and what they expect.

1.3 Standard Interface

All instruments in the system should respond to to at least two types of message: note and control. The OSC address patterns they expect these on is `/<name>/note,ii` and `/<name>/control,ii` where `<name>` is the name of the instrument. By default the client will produce these from incoming MIDI note on and control change messages respectively, choosing the instrument from the channel and sending each of the data bytes as the two OSC integer arguments. The server will listen for these messages even if not defined in the file for the instrument and the MIDI instrument class produces default messages of a note on or control changeon channel 1, with the two OSC arguments making up the two data bytes.