

CARNEGIE MELLON UNIVERSITY

MASTER OF SCIENCE IN MUSIC AND TECHNOLOGY

---

**A Proposal for a Music Player  
for the Human Computer Music Performance Project**

---

*Author:*

Dalong CHENG

*Supervisor:*

Roger DANNENBERG

Richard STERN

Richard RANDALL

October 31, 2012

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Software Architecture . . . . .	1
1.2	HCMP Midi Player Mode . . . . .	2
<b>2</b>	<b>General Design</b>	<b>2</b>
2.1	HCMP Midi Player GUI Design . . . . .	3
2.1.1	Midi Keyboard and Data Display . . . . .	3
2.2	HCMP Midi Player Library . . . . .	3
2.3	HCMP Midi Player API . . . . .	4
<b>3</b>	<b>Implementation</b>	<b>4</b>
3.1	Performer Thread . . . . .	5
<b>4</b>	<b>Future Work</b>	<b>5</b>

# 1 Introduction

The goal of the Human Computer Music Performance (HCMP) [1] project is to create an autonomous “artificial performer” with the ability of a human-level musical performance. An important component of the HCMP project is to develop a player component, which can flexibly adjust and respond to changes in a music signal. Figure 1 illustrates the role of the Player in the HCMP project. The Player will receive control messages from the conductor and scheduler during the performance. In my master’s project, I will design, implement and extend the HCMP midi player for the HCMP project.

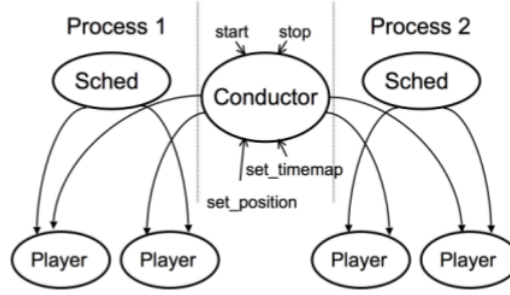


Figure 1: Architecture of HCMP

## 1.1 Software Architecture

In this project, I will use a classic client and server model for the overall architecture. During the performance, the Player will act as the server for the conductor component, which is constantly receiving control messages and responding accordingly.

Internally, the Player will have two threads, with one thread for GUI interactive control (control thread) and the other thread for performing music data (performer thread). The two threads will communicate with each other through a shared message queue, we can assume the message queue is large enough to avoid blocks for both caller and callee threads. The performer thread will handle time critical operations, and there will be a timer setup before this thread is created. The goal of the timer is to wake up the performer thread periodically. Everytime the performer thread’s timer callback function is invoked, it will check the message

queue and process any command from the control thread. Figure 2 illustrates the overall structure of the Player.

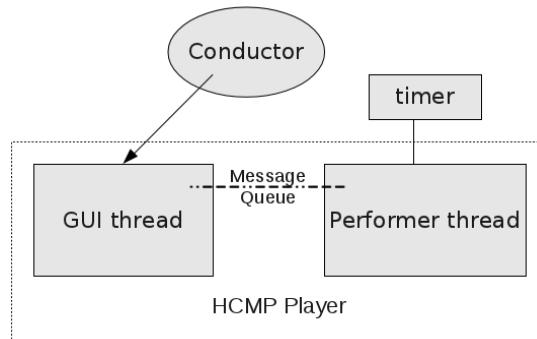


Figure 2: Architecture of the HCMP Midi Player

## 1.2 HCMP Midi Player Mode

The Player will have two modes, stand-alone mode and connection mode. The user is able to switch between the two. In stand-alone mode, the Player is similar to a standard midi player, which can play midi files and set various playback parameters. All user operations will go through the GUI to control. In connection mode the Player will use a predefined protocol to communicate with the conductor and all the control messages will be received from a remote conductor. In connection mode the GUI will use another set of function to map the user operation and some functions of the Player will be controlled by the remote conductor. The control thread of the Player will act as a “proxy”: on one side, it will receive the message from the conductor, on the other side, it will issue a new control message to the performer thread.

## 2 General Design

In this part, I describe the design idea behind some of the features of the Player.

## 2.1 HCMP Midi Player GUI Design

### 2.1.1 Midi Keyboard and Data Display

The Player will contain a virtual keyboard and midi data window. The virtual keyboard will have parameters like channel, tempo, etc for the user to set. It will also be integrated with a midi data window to improve the data visualization effect. After loading a midi file, the midi data window will map each midi note to a key on the midi keyboard. When playing the midi file, the keyboard will highlight each midi note, the midi data window will scroll to the current message position.

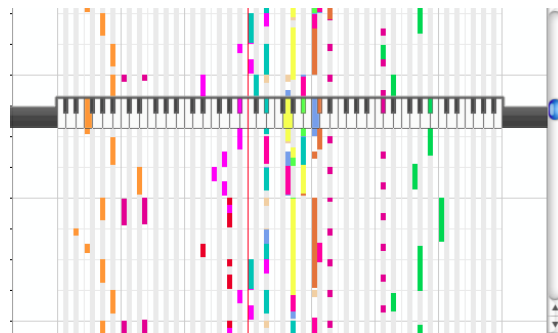


Figure 3: Midi data display integrated with vitural keyboard

## 2.2 HCMP Midi Player Library

The Player will also generate a configuration file to manage and save configurations set by the user. The Player has a simple to use midi library to manage recently played midi information. Figure 4 is an ideal interface for the midi library.

Search...	Played	Artist	Title	Album	Year	Duration	Rating	Genre	Type	Track #	Date	
Library	<input checked="" type="checkbox"/>	(2)	Emmy Verhey...	Sonata No. 5 ...	The 99 Most ...	2010	01:20	★★★★	Classical	mp3	23	3/16/12
Auto DJ	<input checked="" type="checkbox"/>	(2)	Emmy Verhey...	Sonata No. 5 ...	The 99 Most ...	2010	06:32	★★★★	Classical	mp3	24	3/16/12
Playlists	<input type="checkbox"/>	(0)	Emmy Verhey...	Concerto in D...	The 99 Most ...	2010	18:55	★★★★	Classical	mp3	18	3/16/12
Crates	<input type="checkbox"/>	(0)	Emmy Verhey...	Concerto in D...	The 99 Most ...	2010	06:55	★★★★	Classical	mp3	19	3/16/12
Browse	<input type="checkbox"/>	(0)	Emmy Verhey...	Concerto in D...	The 99 Most ...	2010	10:10	★★★★	Classical	mp3	20	3/16/12
Analyze	<input type="checkbox"/>	(0)	Emmy Verhey...	Concerto No. ...	The 99 Most ...	2010	03:52	★★★★	Classical	mp3	25	3/16/12
Rhythmbox	<input type="checkbox"/>	(0)	Emmy Verhey...	Concerto No. ...	The 99 Most ...	2010	06:30	★★★★	Classical	mp3	26	3/16/12
	<input type="checkbox"/>	(0)	Emmy Verhey...	Concerto No. ...	The 99 Most ...	2010	03:59	★★★★	Classical	mp3	27	3/16/12
	<input type="checkbox"/>	(0)	Emmy Verhey...	Concerto No. ...	The 99 Most ...	2010	09:45	★★★★	Classical	mp3	32	3/16/12
	<input type="checkbox"/>	(0)	Emmy Verhey...	Concerto No. ...	The 99 Most ...	2010	07:41	★★★★	Classical	mp3	33	3/16/12
	<input type="checkbox"/>	(0)	Jaime Laredo ...	Concerto in E ...	The 99 Most ...	2010	13:07	★★★★	Classical	mp3	14	3/16/12
	<input type="checkbox"/>	(0)	Jaime Laredo ...	Concerto in E ...	The 99 Most ...	2010	07:59	★★★★	Classical	mp3	15	3/16/12
	<input type="checkbox"/>	(0)	Jaime Laredo ...	Concerto in E ...	The 99 Most ...	2010	06:58	★★★★	Classical	mp3	16	3/16/12

Figure 4: HCMP Midi player library

## 2.3 HCMP Midi Player API

In this section, I list some of the core APIs of the HCMP Midi Player. These APIs will be called by the GUI of the HCMP Player.

Music play related APIs

- `play` - play midi message
- `reset` - reset all the player's settings to default value
- `pause` - store current play information
- `set_position` - set play position to the given parameter

Player setting related APIs

- `set_track` - set track for the player to play
- `set_tempo` - set tempo information
- `set_channel` - set channel information
- `set_device` - set which midi device to send messages to

Connection mode related APIs, which is inherited from Zero MQ [2]

- `play_all` - indicate conductor to play
- `stop_all` - indicate conductor to stop
- `ready` - tell the conductor that the player is ready
- `position` - indicate conductor to set to given position

## 3 Implementation

The project will use Serpent [3] to develop most of the parts, because it has many convenient built in functions for midi messages. For the GUI part, I will use wxWidgets [4]. The first priority of master's project is to deliver a solid and robust midi player. I will design unit test code for each of the functional components during development.

### 3.1 Performer Thread

The performer thread will only receive control messages from the control thread. It will immediately process the message upon receiving it. This thread will be periodically invoked by an external timer, provided by Serpent, and process midi message sent from the control thread. Processing the message is not a time consuming job so the overall overhead of waking up thread will be trivial. Most of work of the performer thread is to maintain a queue of pending midi message and send them at correct time.

## 4 Future Work

This proposal will provide a base for some future work. The Player will be integrated into the HCMP project. Two other features can be added to extend the Player's function. There is a music score display component for the HCMP project. The Player can be used as a default player for music display component. Display with the addition of a midi score follower, the music display could be synchronized to a live keyboard performance and the Player could play an accompaniment part and the user can read the score from the Player's virtual keyboard. Dawen Liang's previous work [5] on music database provide user a convenient way to record, organize and retrieve audio information from various sources. The Player can further integrate this feature to provide a powerful audio library management tool.

## References

- [1] Framework for Coordination and Synchronization of Media, D.Liang, G.Xia and R.Dannenberg, NIME 2011.
- [2] <http://sourceforge.net/p/livedisp/wiki/HCMP>
- [3] <http://www.cs.cmu.edu/music/aura/serpent-info.htm>
- [4] <http://www.wxwidget.org>
- [5] Segmentation, Clustering, and Display in a Personal Music Database for Musicians, G.Xia, D.Liang, R.Dannenberg, ISMIR 2011.