

A Flexible Tool for the Visualization and Manipulation of Musical Mapping Networks

Aaron Henry Krajeski



Department of Music Technology
Schulich School of Music, McGill University
Montreal, Canada

June 2013

A thesis submitted to McGill University in partial fulfillment of the requirements for the degree of Master of Arts.

© 2013 Aaron Henry Krajeski

Abstract

This report describes the use of L^AT_EX to format a thesis. A number of topics are covered: content and organization of the thesis, L^AT_EX macros for controlling the thesis layout, formatting mathematical expressions, generating bibliographic references, importing figures and graphs, generating graphs in MATLAB, and formatting tables. The L^AT_EX macros used to format a thesis (and this document) are described.

Acknowledgments

Acknowledge this, asshole.

Preface

There are some things I should probably pre-face, certainly not reface.

Contents

1	Introduction & Motivation	1
1.1	Context and Motivation	1
1.2	Project Overview	1
1.3	Thesis Overview	1
1.4	Contributions	1
2	Background	2
2.1	Mapping	2
2.2	Interface Design	2
2.2.1	MVC	2
2.3	Visual Design	2
3	Design & Implementation	3
3.1	User Centric Design	3
3.2	Development of a “Modular” Interface	4
3.3	The Model-View-Controller	4
3.3.1	The Model	4
3.3.2	Controller-View Pairs	4
3.4	Graphical Design	4
3.4.1	Typography	4
3.5	Robustness and Responsiveness	4
4	Results & Discussion	5
4.1	Undoing and Redoing in a Collaborative Distributed Environment	5
4.2	Edge Use Cases	5

Contents	v
<hr/>	
4.3 User Feedback	5
4.4 Modular vs Hard-Coded	5
4.4.1 Was the approach successful?	5
4.5 Visualization vs Interaction	5
4.6 Different namespaces	5
5 Conclusions & Future Work	6
5.1 Summary and Conclusions	6
5.2 Future Work	6
References	7

List of Figures

various graphs of response time (discussion) screenshot of drawing screenshot of saving/loading screenshot of main view screenshot of grid view

List of Tables

List of Acronyms

IDMIL	Input Devices for Musical Interaction Laboratory
MVC	Model View Controller
DMI	Digital Musical Instrument
OSC	Open Sound Control
GUI	Graphical User Interface

Chapter 1

Introduction & Motivation

1.1 Context and Motivation

1.2 Project Overview

1.3 Thesis Overview

1.4 Contributions

Chapter 2

Background

2.1 Mapping

Libmapper, vizmapper

2.2 Interface Design

2.2.1 MVC

2.3 Visual Design

Chapter 3

Design & Implementation

Development of a graphical user interface for libmapper creates a unique challenge. Obviously such an interface is a practical tool, and should function as such, yet it also must work in concert with DMIs which are inherently designed for abstract and creative use. For the purposes of this project, the assumed solution to this innate paradox is to provide the user with multiple independent modes of control. This assumption was made based on experiences with prior user interfaces for libmapper (vizmapper, max mapperGUI): for each interface users reported excellent functionality for certain use cases, and poor functionality for others. Libmapper itself is an extremely flexible API that makes few assumptions as to the network of devices and signals, nor how they are being mapped. It is fitting that a GUI for libmapper would be equally as flexible. In lieu of a single perfect solution for network visualization and interactivity, providing users with various independent solutions provided a good compromise.

3.1 User Centric Design

use cases

3.2 Development of a “Modular” Interface

3.3 The Model-View-Controller

Because a modular design is desired, the Model-View-Controller (MVC) metaphor for structuring software applications as described in [KrasnerPope88] was used as a general framework for structuring the application. In fact, the whole scale swapping in and out of independent visual modes can be thought of as a quintessential implementation of MVC.

3.3.1 The Model

The model consists of an abstract copy of the network, residing on the local machine. Independent views can consult this data, but cannot directly modify it.

3.3.2 Controller-View Pairs

3.4 Graphical Design

wiggly arrows

3.4.1 Typography

3.5 Robustness and Responsiveness

speed tests

Chapter 4

Results & Discussion

4.1 Undoing and Redoing in a Collaborative Distributed Environment

4.2 Edge Use Cases

4.3 User Feedback

4.4 Modular vs Hard-Coded

4.4.1 Was the approach successful?

Are sections graphically unified? (Is this even necessary?)

4.5 Visualization vs Interaction

4.6 Different namespaces

Chapter 5

Conclusions & Future Work

5.1 Summary and Conclusions

5.2 Future Work

References

- [1] P. N. Juslin and D. Västfjäll, “Emotional responses to music: The need to consider underlying mechanisms,” *Behavioral and Brain Sciences*, vol. 31, no. 5, pp. 559–621, 2008.