Copyright is owned by the Author of the thesis. Permission is given for a copy to be downloaded by an individual for the purpose of research and private study only. The thesis may not be reproduced elsewhere without the permission of the Author.

CONTEMPORARY APPROACHES TO LIVE COMPUTER MUSIC: THE EVOLUTION OF THE PERFORMER COMPOSER

BY

OWEN SKIPPER VALLIS

A thesis
submitted to the Victoria University of
Wellington
in fulfillment of the requirements for the degree
of
Doctor of Philosophy

Victoria University of Wellington

2013

Supervisory Committee

Dr. Ajay Kapur (New Zealand School of Music)

Supervisor

Dr. Dugal McKinnon (New Zealand School of Music)

Co-Supervisor

© OWEN VALLIS, 2013

NEW ZEALAND SCHOOL OF MUSIC

Abstract

This thesis examines contemporary approaches to live computer music, and the impact they have on the evolution of the composer performer. How do online resources and communities impact the design and creation of new musical interfaces used for live computer music? Can we use machine learning to augment and extend the expressive potential of a single live musician? How can these tools be integrated into ensembles of computer musicians? Given these tools, can we understand the computer musician within the traditional context of acoustic instrumentalists, or do we require new concepts and taxonomies? Lastly, how do audiences perceive and understand these new technologies, and what does this mean for the connection between musician and audience?

The focus of the research presented in this dissertation examines the application of current computing technology towards furthering the field of live computer music. This field is diverse and rich, with individual live computer musicians developing custom instruments and unique modes of performance. This diversity leads to the development of new models of performance, and the evolution of established approaches to live instrumental music.

This research was conducted in several parts. The first section examines how online communities are iteratively developing interfaces for computer music. Several case studies are presented as examples of how online communities are helping to drive new developments in musical interface design.

This thesis also presents research into designing real-time interactive systems capable of creating a virtual model of an existing performer, that then allows the model's output to be contextualized by a second performer's live input. These systems allow for a solo live musician's single action to be multiplied into many different, but contextually dependent, actions.

Additionally, this thesis looks at contemporary approaches to local networked ensembles, the concept of shared social instruments, and the ways in which the previously described research can be used in these ensembles.

The primary contributions of these efforts include (1) the development of several new open-source interfaces for live computer music, and the examination of the effect that online communities have on the evolution of musical interfaces; (2) the development of a novel approach to search based interactive musical agents; (3) examining how networked music ensembles can provided new forms of shared social instruments.

Acknowledgements

The author wishes to express sincere appreciation to all those who have helped with the realization of this thesis. The work that follows is the culmination of the last ten years of my life, and was made possible by the many artists, colleagues, and friends who have inspired and helped me along the way.

Ajay Kapur, thank you first and foremost for your exceptional tutelage in not only my research, but also my life. I would not be here today if it was not for your insight, advice, mentorship, and motivation. You have the amazing gift of seeing the potential in those around you, and the genius to help us realize that potential.

Jordan Hochenbaum, thank you for all the projects, music, and art that we have worked on over the last few years, for all the incredible projects we will work on in the coming years, and most of all for being an incredible friend. Our work as Flipmu has been a source of many of the ideas in this thesis, and I look forward to all the coding, installation, music, and general madness that are to come. You are an incredible musician, amazing artist, and constant source of inspiration.

Many thanks to Dugal McKinnon, Nick Collins, Michael Norris, Martijn Zwartjes, Jim Murphy, Brad Hill, Tim Exley, Jason Edwards, Michael Darling, and Johnny McClymont for their help, ideas, and advice along the way. I would not have been able to complete this thesis without you sharing your expertise and thoughts.

Much of this thesis is the direct result of my experiences performing with other musicians in live computer music ensembles. Thank you to The KarmetiK Machine Orchestra; whom's broad musical vision allowed me to perform both with incredible musicians, and a shared robotic instrument. Thank you to Trimpin, Curtis Bahn, and Tomie Hahn for sharing your incredible work. Your

ideas have helped to inspire my own efforts. Thank you to Jeffery Lufkin for being a part of the genesis for many of the ideas found in this thesis.

Lastly, thank you to my family for their support throughout this journey. To my parents for their encouragement and support; to my incredible and loving wife Liv, for being my source of strength and my muse; and to Rory, for being an inspiration for my writing, and helping me through many challenging discussions, thoughts, and edits.

For any of my friends who I have most regrettably left out of these acknowledgments, please know that you have my deepest gratitude for your help.

TABLE OF CONTENTS

CHAPTER 1 INTRODUCTION 1

1.1 INTERACTION CONTEXTS 2

1.2 PERSONAL MOTIVATIONS 5

1.3 THESIS OVERVIEW 7

CHAPTER 2 COMMUNITY BASED DESIGN: ITERATIVE MUSICAL INTERFACE DEVELOPMENT 11

2.1 GOALS AND MOTIVATION 12

2.2 INTERFACE DEVELOPMENT PRIOR TO ONLINE COMMUNITIES 13

2.3 COMMUNITY BASED DESIGN 15

2.4 ITERATIVE DEVELOPMENT 16

2.5 BACKGROUND CASE STUDIES ON THE MONOME 16

2.5.1 Monome 17

2.5.2 The computer musican as digital luthier 22

2.6 NEW WORK: ARDUINOME AND CHRONOME 22

2.7 MONOME, TENORI-ON COMPARISON 28

2.8 DISCUSSION 29

CHAPTER 3 ARMY OF ME: AUTONOMOUS AGENTS AND THE SOLO PERFORMER 33

3.1 GOALS AND MOTIVATION 34

3.2 BACKGROUND 35

3.2.1 Twentieth century composers 36

3.2.2 Computer aided algorithimic composition 38

3.2.3 Interactive musical agents 40

3.2.4 Contemporary systems 41

3.3 DEFINE THE CHALLENGE 42

3.4 IMPLEMENTATION: LIVE PERFORMANCE SYSTEMS 43

3.4.1 Search based systems 44

3.4.2 Considerations for use with improvisation 45

3.4.3 S2MP: a similarity matching algorithm 46

3.4.4 Training the system, and linking controller data 47

3.4.5 Implementation of the system for use in performance 49

3.4.6 Plugin design 53

3.4.7 Analysis 55
3.4.8 Challenges with using search based systems 64
3.5 Discussion 65
3.5.1 Architecture of an interactive musical agent 65
3.5.2 The army of me 67

Chapter 4 The art of communication: shared instruments and networked musical ensembles 69
4.1 Background 72
4.2 Physicality in computer music peformance, and extending shared control to musical robotics 74
4.3 Musical Robotics and The Karmetik Machine Orchestra 76
4.4 Compositions and performances 78
4.4.1 January 27, 2010 REDCAT - The Machine Orchestra 78
4.4.2 August 14, 2010 – Karmetik Collective 82
4.4.3 April 12, 2012 REDCAT – Samsara The Machine Orchestra 84
4.5 Discussion 86

Chapter 5 Conclusion 89
5.1.2 Improvisation in live computer music 94
5.2 Contributions 96
5.2.1 Online Community based iterative design and the Chronome 96
5.2.2 S2MP and an interactive system for continuous control 97
5.2.3 Shared social musical robotics 98
5.3 Future work and Philosophy 98
5.3.1 Bridging the gap between performer and audience 99
5.3.2 Final thoughts 102

Appendix A Related publications 105

Appendix B Chronome technical files 107

Appendix C Comparative survey of local network ensembles and solo live computer music 121

Appendix D Probabilities and Markov Models 131

Appendix E Search-based algorithms 145

Appendix F Regression systems 151

Appendix G Comparisons and requirements 157

BIBLIOGRAPHY 167

LIST OF FIGURES

Figure 1: Tools used in the evolving roles of the performer composer 7
Figure 2: Overview of the thesis layout 8
Figure 3: Iterations of the Monome 17
Figure 4: Commercially developed Native Instruments Maschine, Akai APC 40, Novation Launchpad, and Ableton PUSH 21
Figure 5: The first Arduinome. Built using Monome 40h buttons 24
Figure 6: Chronome prototype 26
Figure 7: BrickTable version I, II, and III 27
Figure 8: Monome left and Tenori-On right 28
Figure 9: Interactive musical agent for modeling continuous control data from musical interfaces 35
Figure 10: Controller data from both performers stored as 2-D and flattened 1-D collections 47
Figure 11 Search algorithm training algorithm. The two performers are sampled by the plugin, and performer B's current item set is linked with Performer A's previous item set. 48
Figure 12: Stored performances for S2MP algorithm. Sequential item sets for a performance are stored as $Seq_{2(n)}$. Each j^{th} item set represents both Performer A and B, and has a MIDI buffer associated with it. 49
Figure 13: Constraining the S2MP search to current position within a bar. This represents a good compromise between efficiency and search as it cuts the number of compares and is likely to return good matches. 50
Figure 14: Mapping score matrix. The intersection is taken between every i^{th} item set in Seq_1 and the j_{th} item set in the current sub-sequence from $Seq_{2(n)}$. 51
Figure 15: Performance diagram of search algorithm. Sampling Performer A along with the current agent output generates the input sequence. This is then compared against sub-sequences within the database, providing similarity scores. The MIDI buffer is taken from the sub-sequence with the highest similarity score, and then appended to the plugin's MIDI output. 53
Figure 16: Continuous Control S2MP plugin 54
Figure 17: Parsing MIDI buffers into sample windows 54

Figure 18: Routing setup for S2MP training in Ableton Live, and MIDI CC training data 56
Figure 19: Target CC sequence (top) vs. Output sequence (bottom). The output sequence didn't match the target sequence in the first bar, but otherwise was a perfect match. 56
Figure 20: Initial S2MP plugin test - The training sequence length was increased from one to four bars. 57
Figure 21: Similarity percentage between output sequence and target sequence based on Mapping / Order weighting 59
Figure 22: Mapping / Order - Average distance of output sequence from the target sequence 60
Figure 23: Factor increase of discontinuities between output and target sequence 61
Figure 24: Similarity percentage between output sequence and target sequence based on number of item sets in the input sequence, and number of trained transitions 62
Figure 25: Increasing numbers of item sets - average distance of output sequence from the target sequence for input sequences 63
Figure 26: Factor increase of number of discontinuous matches by sequence size 64
Figure 27: Different design approaches for interactive musical agents. Clockwise from top: (A) All inputs affecting the model's output; (B) The model is only affected by itself, and live input is applied as a fitness function; (C) Inputs are split into simpler individual models, all acting independent of each other. 66
Figure 28: Network topology of The Machine Orchestra ensemble 70
Figure 29: The League of Automatic Composers 1980 72
Figure 30: View of the marimbas from the musical robot Tammy 76
Figure 31: The Machine Orchestra at REDCAT 2010 79
Figure 32: The Machine Orchestra performing Samsara 2012 85
Figure 33: The Monome can be both highly programmable or immediately usable .. 90
Figure 34 Preference performing solo computer music vs. networked ensembles .. 122
Figure 35: A Markov Model representing the transition probabilities for the set of notes C through A. The notes on the left represent the source states, and the notes along the top are destination states. The values in the matrix are the transition probabilities for moving from a source state to a destination state. 133
Figure 36: Algorithm for determining the next destination state in a Markov Model given a row of transition probabilities 134

Figure 37: 2nd order Markov Model 135
Figure 38: 2nd order Markov Model shown as 1st order Markov Model 135
Figure 39: The description of a transition probability, containing the Source State, Destination State, and the number of times the transition has been observed during training 137
Figure 40: Markov source states stored as tree structure. This allows for searching variable length source state sequences. 137
Figure 41: Longest matching source state sequence for input sequence F G D C ... 138
Figure 42: Longest matching source state sequence for input sequence G A D C .. 138
Figure 43: Training diagram of Markov Model 141
Figure 44: Performance diagram of Markov Model 142
Figure 45: Input to Markov Model that does not match any previously seen source state 143
Figure 46: Matching sequences return the next stored state form the database..... 146
Figure 47: Matching sequences return values from a second linked sequence..... 147
Figure 48: Matrix with ideal rating scores comparing input sequence along the columns, and scored sequence down the rows 148
Figure 49: Insertion test: this tests for notes played by the performer that are not in the original score 149
Figure 50: Deletion test: this tests for notes from the score that are skipped over by the performer 149
Figure 51: Substitution test: this tests for notes substituted by the performer..... 149
Figure 52: Repetition test: this looks for notes from the score that are repeated by the performer 149
Figure 53: Constrained search to speed up sequence comparison 150
Figure 54: Basic linear regression 152
Figure 55 Similarity between linear regression and basic sequential ANN 152
Figure 56 Diagram of multi-variate linear regression model 154

LIST OF TABLES

Table 1 Reasons for solo or group performance preference..... 124
Table 2 Descriptions of each musician's roll within the ensemble..... 125
Table 3 Description of new performance modes afforded by computer ensembles 127
Table 4 Descriptions of the challenges of performing in computer ensembles..... 129
Table 5 Overview of algorithms for designing interactive musical agents..... 164