2013

Interactive Musical Partner: A System for Human/Computer Duo Improvisations

Jeffrey Vernon Albert
Louisiana State University and Agricultural and Mechanical College

INTERACTIVE MUSICAL PARTNER:
A SYSTEM FOR HUMAN/COMPUTER DUO IMPROVISATIONS

A Dissertation

Submitted to the Graduate Faculty of the
Louisiana State University and
Agricultural and Mechanical College
in partial fulfillment of the
Requirements for the degree of
Doctor of Philosophy

in

The School of Music

by
Jeffrey V. Albert
B.M., Loyola University New Orleans, 1993
M.M., University of New Orleans, 2000
May 2013

ACKNOWLEDGEMENTS

This project would have been impossible without the support and encouragement of my family. To Jennifer, Jessica, Emily and Blake: thank you, I love you, and I could not have done this without you.

Professors Stephen David Beck and Jesse Allison have been instrumental in guiding this research and helping me to develop as a technologist and programmer in ways that have pushed my development as an artist. My colleagues and fellow students in the EMDM program have always been supportive and encouraging, and I would like to specifically thank Nick Hwang and J. Corey Knoll who have been good friends, valuable sounding boards, and generous collaborators.

A number of truly world-class musicians have helped me by participating in interviews, playing duets, and beta testing the software. I would like to thank: Jeb Bishop, Dave Rempis, Steve Swell, Dan Oestreicher, Brad Walker, Rick Trolsen, Mark McGrain, Janna Saslaw, and Ray Moore (who also provided some of the fine Brazilian coffee that fueled the final days of writing).

TABLE OF CONTENTS

ABSTRACT

This research is centered on the creation of a computer program that will make music with a human improviser. This Interactive Musical Partner (IMP) is designed for duo improvisations, with one human improviser and one instance of IMP, focusing on a freely improvised duo aesthetic. IMP has Musical Personality Settings (MPS) that can be set prior to performance, and these MPS guide the way IMP responds to musical input from the human. The MPS also govern the probability of particular outcomes from IMP's creative algorithms. IMP uses audio data feature extraction methods to listen to the human partner, and react to, or ignore, the human's musical input, based on the current MPS.

This course of research presents a number of problems. Parameters for the Musical Personality Settings (MPS) must be defined, and then those parameters must be mapped to extractable audio features. A system for musical decision-making and reaction/interaction (action/interaction module) must be in place, and a synthesis module that allows for MPS control must be deployed.

Designing a program intended to play with an improviser, and then improvising with that program has caused me to assess every aspect of my practice as an improviser. Not only has this research expanded my understanding of the technologies involved and made me a better technologist, but striving to get the technology to be musical has made me look at all sides of the music I make, resulting in a better improvising artist.

CHAPTER 1: INTRODUCTION

> At the uninterrogated core of powerful new forms of computer interactivity that challenge traditional conceptions of human identity, we find the primordial human practice of improvisation. - George Lewis[1]

Improvisation is present in virtually every musical tradition in the world[2], thus it is reasonable that it should also be present in the computer music tradition. Improvisers and electronic music experimentalists often have shared aesthetic values, and the intersection of improvisation and electronics came early in the timeline of live performance with electronic instruments. Musica Elettronica Viva, or MEV, was formed in 1966 and their basic performance practice centered on group improvisation with electronic instruments of the member's own construction.[3] There is also a strong history of electronic musicians performing and recording with jazz-identified improvisers. The collaborations between Richard Teitelbaum and Anthony Braxton[4], and Jon Appleton and Don Cherry[5] stand out in this area. These are all examples of more traditional paradigms of improvised interactions, which also include electronic instruments. This paradigm of improvised interaction involves a number of humans interacting with each other. The addition of electronic

---

[1] George Lewis, "Mobilitas Animi: Improvising Technologies, Intending Chance," *Parallax* 13, no. 4 (2007).

[2] Gabriel Solis and Bruno Nettl, eds., *Musical Improvisation: Art, Education, and Society* (Urbana and Chicago: University of Illinois Press, 2009), 3.

[3] Joel Chadabe, *Electric Sound: The Past and Promise of Electronic Music* (Upper Saddle River, NJ: Prentice Hall, 1997), 103.

[4] Anthony Braxton and Richard Teitelbaum, *Silence / Time Zones* ([S.l.]: Black Lion).

[5] Jon H. Appleton and Don Cherry, *Human Music* (New York: Flying Dutchman).

instruments did not change the basic nature of the interactions; it simply changed the timbres of the sounds.

The advent of personal computers capable of musical performance allowed for new models of interaction between musicians and their electronics. The group known as The Hub modeled their musical interactions on computer network infrastructures. The network became the instrument with musical responsibilities being shared by the group members in new ways.[6] The "computer that processes performer input" is another model of musician/computer interaction. A good example of this type of interaction is Richard Teitelbaum's *Concert Grosso for Human Concertino and Robotic Ripieno* (1985). The composer describes the system: "material played live by the composer-performer on one piano keyboard was instantly read into a computer memory where it was processed (delayed, overlaid, looped, transposed, speed-changed, randomized, etc.) and simultaneously played out."[7] Performer input was mostly in MIDI format in the earlier works of this type, but more recent works have moved towards using raw audio input.[8]

Teitelbaum uses the term "composer-performer" and this research deals directly with improvisation, so at some point the line that separates composition and improvisation must be addressed. In his keynote address at ImproTech Paris/NY 2012[9], George Lewis stated that he

---

6 Chadabe, 297.

7 Richard Teitelbaum, Anthony Braxton, and George Lewis, *Concerto Grosso for Human Concertino and Robotic Ripieno* (Therwil, Switzerland: Hat Hut Records).

8 Robert Rowe, "Personal Effects: Weaning Interactive Systems from Midi," in *Proceedings of the Spark Festival 2005* (Minneapolis, MN: 2005).

9 IRCAM, "Improtech Paris/Ny 2012: Improvisation & Technology" http://repmus.ircam.fr/improtechpny (accessed May 27, 2012).

does not feel that finding that line is a rewarding endeavor.[10] Nonetheless briefly defining each will help to clarify the goals of this research, and the methods chosen to reach those goals. A basic difference between composition and improvisation is that in composition the creation time and performance time are not on the same scale, whereas in improvisation creation time and performance time are the same. Improvisation, according to Bill Evans, is "making one minutes music in one minute's time, where a composer might take three weeks to make one minute's music."[11]

Computer algorithms used to compose music are an important piece of the history that has led us to a situation in which computers can be active participants in freely improvised experiences. This oft-belabored dichotomy between composition and improvisation becomes meaningless when we look at single-agent computerized musical creation systems. Whether the computer ran its algorithm on stage while the sound was being made or at a previous time to be recorded and replayed at the moment of performance is irrelevant, if the computer is the only agent involved. The addition of other musical agents requires interaction, or even dialog, between those agents in that same creation/performance time, and significantly complicates the requirements for musical behavior on the part of the computer.

[10] For an in depth exploration of this topic see:
Bruce Ellis Benson, *The Improvisation of Musical Dialogue: A Phenomenology of Music* (New York: Cambridge University Press, 2003).

[11] Bill Evans, Harry Evans, and Steve Allen, *The Universal Mind of Bill Evans: The Creative Mind and Self-Teaching* (New York, N.Y.: Rhapsody Films).

Computers as algorithmic music compositional tools have a rich history starting with the work of Lejaren Hiller and continuing beyond the work of David Cope[12]. Hiller's work in computer generated composition was some of the first, and Cope's work is possibly the most widely known, having been featured a number of times in mainstream media like National Public Radio[13] and the New York Times.[14] Each of these men devised a system in which the computer would generate musical output based on a set of rules. Computers are very good at following rules, so algorithmic composition is a fairly well developed field. The work of Hiller, Cope, and others has put the focus on using algorithms to compose music to be performed by humans. The output of the algorithm is sheet music.

The rise of a generation of computer languages focused on sound synthesis has led to a closer integration of the algorithmic generation of instructions and the actual synthesis of sounds. SuperCollider, ChucK, Max, and Pd all provide environments in which one can build algorithms which will generate events and also execute the sound synthesis. The development of these sorts of integrated algorithmic generation/synthesis systems is an important step in allowing the creation of the system to be described here.

Improvisation with multiple agents involves listening and making musical decisions based on that listening. When those improvising agents are humans, those musical decisions are based on aesthetic judgments which are grounded in personal experience and taste, so the

---

[12] James Harley, "Computational Approaches to Composition of Notated Instrumental Music: Xenakis and the Other Pioneers," in *The Oxford Handbook of Computer Music*, ed. Roger T. Dean(Oxford: Oxford University Press, 2009), 113-115, 125-4.

[13] "Virtual Composer Creates New Music" http://www.npr.org/templates/story/story.php?storyId=113719483 (accessed May 22, 2012).

[14] George Johnson, "Undiscovered Bach? No, a Computer Wrote It," *The New York Times*November 11, 1997.

problem for a programmer wishing to create a computer agent that will participate in an improvised musical experience becomes how to program experience and taste. Experience can be translated into rules that the computer can follow. Machine learning algorithms can allow us to “teach” the computer to “hear” as anyone of us might hear, or more accurately to associate a given audio feature with a given musical parameter. Although the concept of taste is a bit more complicated, if we can program a computer to compose in a fashion that we find tasteful, it follows that we should be able to program a computer to react to musical input in a way we find tasteful. This is the crux of this research.

This research is centered on the creation of a computer program that will make music with a human improviser. This Interactive Musical Partner (IMP) is designed for duo improvisations, with one human improviser and one instance of IMP, focusing on a freely improvised duo aesthetic. IMP has Musical Personality Settings (MPS) that can be set prior to performance, and these MPS guide the way IMP responds to musical input from the human. The MPS also govern the probability of particular outcomes from IMP’s creative algorithms. The IMP uses audio data feature extraction methods to listen to the human partner, and react to, or ignore, the human’s musical input, based on the current MPS.

This course of research presents a number of problems. Parameters for the Musical Personality Settings (MPS) must be defined, and then extractable audio features must be identified and correlated with the MPS. A system for musical decision-making and reaction/interaction must be in place, and a synthesis module that allows for MPS control must be deployed.

The bulk of this research is focused on the development of the MPS, and the correlation of the parameters of the MPS with extractable audio features. The parameters of the MPS have

been determined initially by my experience and intuition as it relates to musical parameters that are important to dialogic improvised communication. These decisions have also been informed by interviews that I conducted with other respected improvisers. These parameters were connected to extractable audio features by coding solo recordings made specifically for this purpose, then audio features were extracted from these same excerpts, and a machine-learning algorithm was trained to connect the extracted audio features to the human coded MPS parameters.

IMP was constructed in the Max 6 programming environment[15], using Tristan Jehan's `analyzer~` external[16] for the audio feature extraction tasks, and Rebecca Fiebrink's Wekinator[17] for the machine-learning implementation. A number of objects from the Tap Tools collection of objects and extensions for Max 6 are also used.[18]

Success in this endeavor is defined as a musically viable experience for both the performer and the audience. As a means to determine the success of IMP, a number of noted improvisers have performed with IMP, and completed a survey about the experience. The following chapters will outline the prior art upon which IMP is based (both aesthetic and technical), the process of development of IMP, an in depth description of the structure of IMP

---

15 http://cycling74.com/products/max/

16 Tristan Jehan, "Max/Msp Externals" http://web.media.mit.edu/~tristan/ (accessed January 27, 2013).

17 Rebecca Fiebrink, Dan Trueman, and Perry Cook, "The Wekinator: Software for Using Machine Learning to Build Real-Time Interactive Systems " http://wekinator.cs.princeton.edu/ (accessed November 29, 2011).

18 Tim Place and Jesse Allison, "Tap Tools" http://74objects.com/taptools/ (accessed March 19, 2013).

and the reasoning behind those structural decisions, instructions for the use of IMP, and an analysis of the success of the research to this point.

CHAPTER 2: PERSONIFICATION, INTENTION, AND MEANING

In the following pages, many words are used to describe the events initiated by IMP. There is writing about IMP's "behavior" and the way that IMP "reacts" to particular musical input. It is very easy to discuss IMP and IMP's actions in ways that personify the program. One might be temped to say that IMP is improvising. Even the way I refer to IMP (by a singular name) encourages this sort of personification.

IMP is not a being. IMP has no intention. IMP assigns no meaning to the sounds it creates. IMP does not improvise. The humans who play with IMP improvise. IMP just follows instructions.

Improvisation is often studied as much as a social practice as it is a musical practice.[19] This sort of study imbues improvisation with deep meaning. The practice of musical improvisation has also been viewed as politically and socially symbolic.[20] The practitioners of improvisation often speak of the profound personal expression available to them through the act of improvising. I will refrain from describing what IMP does as improvisation, because what IMP does lacks the meaning that has come to be associated with the act of improvisation.

The philosopher Edmund Husserl's ideas on intentionality can be summarized as, "[t]he intentional content of an intentional event is the way in which the subject thinks about or

---

[19] One such example is the journal *Critical Studies in Improvisation/Études critique en improvisation*, that is focused on improvisation, community and social practice. http://www.criticalimprov.com/

[20]George E. Lewis, "Living with Creative Machines: An Improviser Reflects," in *Afrogeeks: Beyond the Digital Divide*, ed. Anna Everett and Amber J. Wallace(Santa Barbara, CA: Center for Black Studies Research, University of California, 2007), 85.

presents to herself the intentional object."[21] When humans improvise we create this intentional content. IMP does not create intent. IMP cannot think about or present itself to the intentional object, thus IMP lacks intent.

It can seem obvious to say that a computer program is not a person, and has no intent, and creates no meaning on its own, but when we start to make computer programs that interact with humans in ways that have until recently been reserved for other humans, it can be tempting to personify that program. However, using the language of personification can make communication easier. To write that "IMP heard the human play the notes C, D, and E, then decided to play those notes as well," is much more concise than writing: "the `analyzer~` object in IMP's listener module received signal from the ADC, and that signal triggered `analyzer~` to indicate three different events, and that the FFT analysis of the signal at the time of the onsets of those events revealed fundamental frequencies that are associated with the musical notes C, D, and E. Those frequency values were stored in the heard frequencies list, and a series of random number generations filtered through a weighted probability mechanism caused values to be sent to an FM synthesis module that would play the notes C, D, and E." IMP's actions are described using personified language at times, but this is done for ease and clarity of communication. The use of this personifying language should not be misconstrued as indicating any belief that IMP has intent or creates meaning on its own, or even that IMP is improvising. IMP is simply executing its algorithm.

Early in the process of defining the shape this research was to take, I was discussing the idea(s) with Jeb Bishop, who is a close friend, valued colleague, world-class improvising

[21] Andrew D. Spear, "Husserl on Intentionality and Intentional Content", Internet Encyclopedia of Philosophy: A Peer-Reviewed Academic Resource http://www.iep.utm.edu/huss-int/#SH1a (accessed February 6, 2013).

musician, and (retired) philosopher. Jeb’s initial response was that I would never be able to program a computer to do what a human does when improvising, meaning that the functions of the human mind can not be reduced to algorithmic logic, a point with which I agree. The goal of this research is to create a program that behaves in a way that improvising musicians, and audiences, can find musically rewarding, not to create an algorithm that functions like the brain of an improvising human. The knowledge gained is as much about exploring the aspects of musical interaction that we find compelling, as it is about developing technical aspects of programming algorithms. IMP can serve as a mirror to reflect the inclinations of performers, and a lamp to light the expectations of listeners. Intention is present, and meaning is created, but that intention and meaning come from the human side of the system.

CHAPTER 3: AESTHETIC FOUNDATIONS

> Our aim is to play together with the greatest possible freedom – which, far from meaning without constraint, actually means to play together with sufficient skill and communication to be able to select proper constraints *in the course of the piece*, rather than being dependent upon precisely chosen ones. – Ann Farber[22]

It is important to understand the aesthetic space that a work of art, or piece of software, strives to inhabit, especially if openness is at the heart of that space. IMP strives to function in the aesthetic lineage of the freely improvised duo. This is a lineage that includes the music called free jazz, and is often made by musicians who have been associated with jazz, even if they do not themselves claim the affiliation.[23] Derek Bailey calls freely improvised music, "an activity that encompasses too many different kinds of players, too many different attitudes to music, too many different concepts of what improvisation is, even, for it all to be subsumed under one name."[24] David Borgo describes the music, "often dubbed 'free improvisation'" as tending to, "devalue the two dimensions that have traditionally dominated music representation – quantized pitch and metered durations – in favor of the microsubtleties of timbral and temporal modification."[25] This is an accurate description of the musical priorities of IMP.

IMP is constructed in a way that forgoes the abstractions of notes and rhythms, and deals with both creation of sounds and analysis of heard sounds in terms of the basic characteristics of

---

22 Daniel Belgrad, *The Culture of Spontaneity: Improvisation and the Arts in Postwar America* (Chicago and London: The University of Chicago Press, 1998), 2.

23 George E. Lewis, *A Power Stronger Than Itself: The Aacm and American Experimental Music* (Chicago and London: The University of Chicago Press, 2008), 98.

24 Derek Bailey, *Improvisation: Its Nature and Practice in Music* (New York: Da Capo Press, 1993), 83.

25 David Borgo, *Sync or Swarm: Improvising Music in a Complex Age* (New York: Continuum International Publishing Group, 2005), 3.

frequency and duration. This frees IMP to make and hear sounds that fall outside of the structures of temperament and tempo, but does not preclude IMP from making sounds that we may hear as having those qualities.

IMP participates in musical interactions that are free from predetermined tonalities, or harmonic or rhythmic structures. That is not to say that a tonality or another structure of some sort cannot develop, but that development will be of the moment. While the human performer and audience may interpret the structure of a tonality or tempo from what they hear, IMP is not "thinking" in those terms. Bailey differentiates between idiomatic and non-idiomatic improvisation, with idiomatic improvisation being that which fits a defined style, such as jazz, or flamenco, or baroque. Non-idiomatic improvisation may be stylized, but is not identified as such, nor defined by style or idiom.[26] IMP is designed to interact in non-idiomatic improvisations.

Appendix 1 contains a list of basis recordings that serve to define the target aesthetic of IMP. Most of these recordings have two human agents playing acoustic instruments, although in some of the cases one of the human agents is playing an electronic instrument. On a few of the recordings there is a computer-based interactive agent. These recordings are examples of the types of sonic interactions that have influenced the design of IMP.

Many of the basis recordings involve two monophonic instruments being played by humans. Some of these duos involve composed material as a starting point for the improvisations. A few also involve polyphonic instruments such as piano or melodic percussion, while some include other percussion, such as drums and cymbals. Since IMP is designed to be one voice in a duo of monophonic participants in which there is no previously composed

[26] Bailey, xi-xii.

material, some of these basis recordings fall outside of the strict definition of what IMP does, but they have other factors that justify their inclusion on the list of basis recordings.

IMP produces sound in a monophonic manner, and listens in a way that is designed to parse monophonic input, thus the examples that include two monophonic participants most closely resemble the types musical textures that IMP will produce with a human partner. The other recordings that use polyphonic and percussion instruments are included because all of the duos between two humans represent the type of interaction that IMP aims to emulate. This is what George Lewis calls dialogic interaction, or interaction in which all of the exchange of information is musical. There is no composition or predetermined idiom to mediate the interaction, only the action of each agent and the reactions to that action. The dialogue consists entirely of the sounds.

Dialogic musical interactions include a number of fundamental ways to react to input. Input can be ignored, meaning one agent is aware of the contributions of the other, but chooses not to acknowledge them. Input can be imitated, or partially imitated. One agent can decide to repeat the musical material recently presented by the other agent, or adopt some aspect of the recently presented material, such as pitch selections, rhythmic relationships, timbral qualities, or dynamic shape. Ignoring and imitating are the two simplest reactions. More complex reactions include complementary action, in which one agent avoids imitation of the other, but reacts in a way that is musically agreeable or harmonious in some way.[27] Oppositional action is another possibility. This differs from ignoring the input from the other agent, in that oppositional action requires intentioned action to create sounds on the opposite end of the spectrum in terms of

[27] In this context “harmonious” is used in its broader definition of “forming a pleasing and consistent whole” (New Oxford American Dictionary), as opposed to its more strict musical meaning which relates to the creation of specific musical harmonies.

timbre, density, rhythm, melody, or some combination of those aspects. The decision-making algorithms in IMP do not strictly follow the processes outlined in this paragraph, but they do allow for the same musical outcomes. These algorithms are explained in 5.5 Creative Algorithm and Musical Memory Systems.

CHAPTER 4: TECHNOLOGICAL PRECEDENTS

Interactive has been a popular word in computer music circles in the first decade of the 21st Century. The Oxford Dictionaries define interactive (with relation to a computer or electronic device) as: "allowing a two-way flow of information between a computer and a computer-user; responding to a user's input." Often the word interactive is mistakenly used in place of reactive. I have attended a number of concerts in which a piece was described as instrument and interactive electronics, when in essence the electronics simply reacted to what the instrumentalist played. For interaction to be present, there must be exchange of information in both (or all) directions: from the human to the computer, and from the computer to the human. This chapter will summarize some important milestones in human/computer musical interaction, starting with the SalMar Construction and CEMS, and continuing through the more improvisation focused works of Richard Teitelbaum, George Lewis, and the OMAX team at IRCAM.

In 1969, the Coordinated Electronic Music Studio (CEMS) system was installed at the State University of New York at Albany.[28] The system was conceived by Joel Chadabe and built by Robert Moog. Chadabe described the system as "the realtime equivalent of algorithmic composition,"[29] meaning that its series of sequencers and oscillators could be programmed to automate an entire composition, yet there were also joysticks that allowed a human to influence the various components of the CEMS. The individual components of CEMS could also influence each other, so there was a complex system of influence between the human and the various components of CEMS. Chadabe describes the experience:

---

[28] Chadabe, 286.

[29] Ibid.

> Because I was sharing control of the music with the sequencers, I was only partially controlling the music, and the music, consequently, contained surprising as well as predictable elements. The surprising elements made me react. The predictable elements made me feel like I was exerting some control. It was like conversing with a clever friend who was never boring but always responsive. I was, in effect, conversing with a musical instrument that seemed to have its own interesting personality.[30]

The CEMS system was one of the early systems that provided a mechanism for the human performer to interact the computer, or influence the algorithm. The human influenced the computer via joystick controllers, and the computer influenced the human via sound. Salvatore Martirano and his team at the University of Illinois completed the SalMar Construction in 1972.[31] The SalMar Construction was an instrument that consisted of hundreds of switches that controlled various aspects of some or all of four different voices that could be in any of four different tuning modes. The switches were touch sensitive, and the instrument "could be both manually and logically driven."[32] Martirano describes performing with the SalMar Construction:

> It was too complex to analyze. But it was possible to predict what sound would result and this caused me to lightly touch or slam a switch as if this had an effect. Control was an illusion. But I was in the loop. I was trading swaps with the logic. I enabled paths. Or better, I steered. It was like driving a bus.[33]

The SalMar Construcion and CEMS added a human element of influence to an otherwise machine controlled compositional process creating interactive systems for music making. Each did this via some sort of tactile control input from the human, with the machine returning musical information to the human via sound. The next step in moving towards improvised interactivity is to have the human influence on the machine be aural as well. One of the pioneers

---

[30] Chadabe, 287.

[31] Ibid, 290.

[32] Ibid,, 289.

[33] Ibid, 291.

of this mode of interaction is Richard Teitelbaum, who has created a number of interactive programs “in which the performer controls the computer’s responses by aspects of his/her musical input.”[34] Teitelbaum composed *Concerto Grosso for Human Concertino and Robotic Ripieno* in 1985, and the premiere featured Teitelbaum on piano, along with Anthony Braxton on saxophone and George Lewis on trombone.[35] The performance involved Lewis and Braxton playing through microphones connected to pitch-to-MIDI converters, which fed through an Apple II to Teitelbaum’s Patch Control System, which also had inputs from Teitelbaum’s piano, and outputs to two robotically controlled pianos, and a number of synthesizers. The Patch Control System was built on Patch Control Language (PCL), which was an object oriented programming environment that had a variety of modules that could perform logic and processing functions.[36] The human performers played the role of the concertino, and the robotic pianos were the ripieno in this 20th Century reimagining of the baroque concerto grosso form.

Teitelbaum’s systems record the musical data that has been played into them, then can replay, transform, vary, and modify those data. These systems are quite complex in the ways in which the musical input from the humans affects the behavior of the program. This complexity is intentional. Teitelbaum’s stated goal is to “[interact] with deeper levels of consciousness… By creating a highly complex set of stimuli and responses to the improvised input, the performer is not quite able to comprehend or keep track of all the reactions he/she is affecting.”[37] The idea is

[34] Richard Teitelbaum, "Improvisation, Computers, and the Unconscious Mind," *Contemporary Music Review* 25, no. 5/6 (2006): 504.

[35] Teitelbaum et al.

[36] Teitelbaum, "Improvisation, Computers, and the Unconscious Mind," 503.

[37] Ibid, 504.

to move the human improviser's sense of wanting to control the program out of the conscious mind and let the musical subconscious take over.

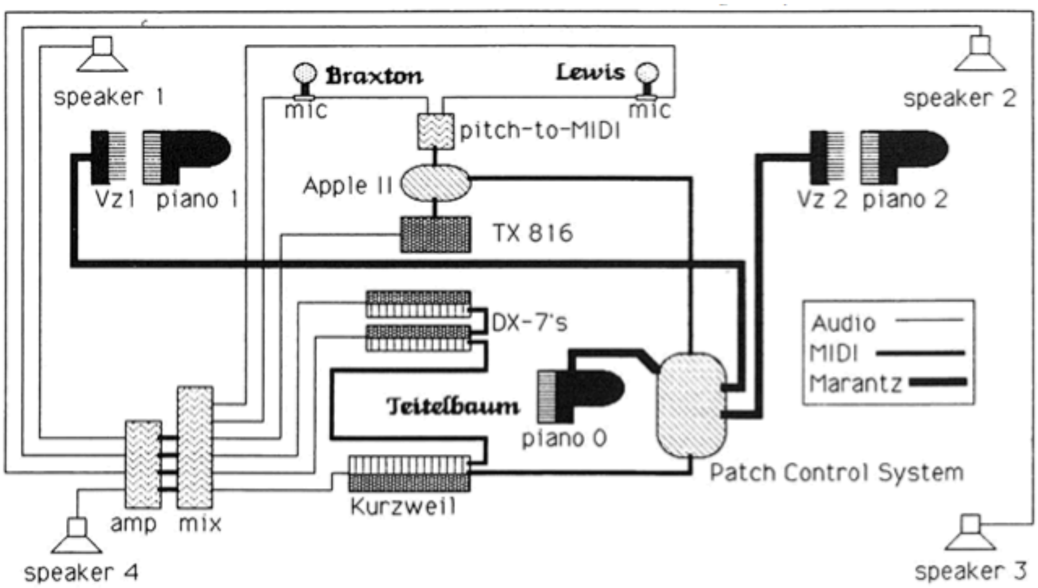

Figure 1. Diagram showing overall configuration for *Concerto Grosso* (Courtesy of Richard Teitelbaum)

Moving from Teitelbaum's idea of subconscious interaction towards more conscious or dialogic interactions leads to a discussion of two systems that get closer to actually improvising with a human, as opposed to accompanying an improvising human, an important distinction.

The first of these systems is George Lewis's *Voyager.*[38] Lewis describes *Voyager's* structure: "a performance of *Voyager* as multiple parallel streams of music generation, emanating from both the computers and the humans – a nonhierarchical, improvisational, subject-subject model of discourse, rather than a stimulus/response setup."[39] Aesthically, *Voyager* is based on the Trans-African aesthetic of multidominance.[40] Multidominance is an idea

[38] George E. Lewis, "Too Many Notes: Computers, Complexity and Culture in "Voyager"," *Leonardo Music Journal* 10, (2000).

[39] Ibid, 34.

that Lewis has adapted from art scholar Robert L. Douglas, which he defined as "the multiple use of colors in intense degrees, or the multiple use of textures, design patterns, or shapes."[41] Lewis applies this theory to African-American improvised music[42], and it can be heard clearly in the many voices that make up *Voyager.* Other musical examples of multidominance are the collective improvisation of traditional jazz bands, the heterophony of certain gospel choirs, and the multi-faceted melodic and rhythmic approach exhibited by groups like the Art Ensemble of Chicago. According to Lewis, "Both the sonic behavior and the program structure of *Voyager* exhibit multidominance in a number of respects… the *Voyager* program is conceived as a set of 64 asynchronously operating single-voice MIDI-controlled "players," all generating music in real time."[43]

A performance of *Voyager* can seem almost orchestral in scope, due to the large number of voices that may be sounding at any given moment. The behavior of these voices is influenced by the musical output of the humans who may be playing with the program, but as with Teitelbaum's work, the system is complex enough that it is impractical for the human to try to "control" the program. The dialogue happens less in the conscious mind than in the musical flow. In his keynote address at ImproTech Paris/NY 2012, Lewis spoke of an improvising machine as a "device that is free with respect to us," continuing to say that our only interface with these devices is via musical dialog. This is the model of discourse, a nonhierarchical,

---

[40] Lewis, "Too Many Notes: Computers, Complexity and Culture in "Voyager"," 33.

[41] Robert L. Douglas, "Formalizing an African-American Aesthetic," *New Art Examiner* 18, (1991).

[42] George E. Lewis, "Purposive Patterning: Jeff Dondaldson, Muhal Richard Abrams, and the Multidominance of Consciousness," *Lenox Avenue: A Journal of Interarts Inquiry* 5, (1999).

[43] Lewis, "Too Many Notes: Computers, Complexity and Culture in "Voyager"," 34.

improvisational subject-subject model, used in *Voyager* and has been adopted for IMP as well. This means that in IMP, like in *Voyager,* both the computer and the human will behave as independent musical agents, and the communication between the two will only be via musical dialog. In addition to the program having the ability to make music free from human input, Lewis also advised me that, "it is important that it makes music *you* like."[44]

IMP will vary from *Voyager* in both technical and aesthetic ways. IMP takes much of its spirit from *Voyager*, especially the ideas about dialogic interaction and the agency of the program. IMP differs from *Voyager* aesthetically in that the aesthetic of multidominance in place in *Voyager* is not present in IMP, but rather it is replaced by an aesthetic of two equal voices, one from the human and one from the program. A primary technological difference is that *Voyager* translates all of the incoming audio into MIDI data, and IMP deals with raw audio features for its listening and synthesis. The primary ways in which IMP differs from Voyager are: IMP has but one voice and is based on a freely improvised duo aesthetic; IMP avoids using the MIDI protocol for analysis or synthesis, and executes those tasks based upon audio feature data.

The other significant, and more recent, work that is directly precedent to this project is OMAX, which is software that has been developed at IRCAM.[45] The OMAX Project Page describes OMAX as, "a software environment which (sic) learns in real-time typical features of a musician's style and plays along with him interactively, giving the flavor of a machine co-improvisation… OMAX reinjects in several different ways the musician's material that has gone through a machine-learning stage, allowing a semantics-level representation of the session and a

[44] Personal conversation. February 11, 2012. Appleton, WI.

[45] IRCAM, "The Omax Project Page" http://omax.ircam.fr/ (accessed May 27, 2012).

smart recombination and transformation of this material in real-time."[46] A significant difference between OMAX and *Voyager*, is that OMAX is conceptually based on re-composition of previously played, and analyzed, material, where *Voyager* generates its own musical material, letting live input have only a dialogic influence upon its output. In early versions of OMAX, the software generated its output based on analysis of a corpus of previously input MIDI information. Later it incorporated real-time input of MIDI information, and current versions use audio information instead of MIDI. The version of OMAX that was demonstrated at ImproTech Paris/NY 2012 by Benjamin Lévy bases its analysis on audio input and re-synthesizes that audio to output the re-composed musical ideas that it has created based on the live input. It sounds like manipulation of recorded samples, but it is actually re-synthesis.

OMAX listens to audio and is focused on re-composition of real-time input. *Voyager* listens via MIDI data and is focused on independent musical generation with only dialogic influence from any real-time input. IMP is a program that listens to audio, uses audio data as the basis for new music generation, and interacts dialogically with a human improviser.

[46] IRCAM, "The Omax Project Page" http://omax.ircam.fr/ (accessed May 27, 2012).

CHAPTER 5: THE SOFTWARE

My goal in the creation of this Interactive Musical Partner (IMP) software was to make a software system that could live up to all three parts of its name. It would have to be interactive, meaning that there is a two-way flow of information between the human performer and IMP. It would have to be musical, meaning that the results are musically rewarding both the performer and to listeners. And finally, it should be a partner, meaning that it is equal parts leader and follower, not always simply accompanying the improvising human, and at the same time, not always requiring the human to accommodate its output.

Two important concept/design decisions were made very early in the process. 1) IMP would be a monophonic participant in its musical settings, and that those settings would fall into the sphere of non-idiomatic improvisation (or free improvisation). This was based on a model of duo improvisations in which I have participated many times, and that experience would guide many of the design and implementation decisions made during this research. 2) IMP would deal with the ways it hears, remembers, and creates musical content with the fewest possible levels of abstraction. Whenever possible pitches are dealt with in terms of frequency and durations in terms of milliseconds. By avoiding "thinking" in terms of note names and note vales, IMP can more easily navigate the spaces outside of tonality and tempo. There are two notable exceptions to this principle, which will be discussed later in this chapter.

One of the initial hopes for IMP was that it would have a musical personality capable of morphing in infinite ways. Very early in the design process it became clear that endless possibilities lead to endless complexity, so IMP has a somewhat more consistent musical character now. It has, however, shown the ability to behave musically with a variety of quite different musical partners. I was hoping to create something that was not simply an algorithmic

extension of my improvising aesthetic, but the more I have played with IMP, the more of myself I hear in it, and it makes sense that my musical sensibilities would find their way into this software of my design.

IMP was programmed in the Max 6 Programming environment, and uses The Wekinator for machine learning implementation. IMP consists of: a synthesis module, a Musical Personality Settings (MPS) module, a frequency decider, a duration decider, a global variation module, and timbral noise analysis module, a listener module, and a number of smaller decider modules. I will use the term creative algorithm to refer to the aspects of the duration decider and frequency decider that control IMP's autonomous output. Each of these components will be described in terms of design and implementation in this chapter.

5.1 SYNTHESIS MODULE

The synthesis module is IMP's voice. This is that section of the software that makes the sounds, and manages the messages sent from the other modules. The frequency decider, duration decider, global variation module, and listener module are all sub-patches of the synthesis module. IMP's voice sounds like computer synthesis, and that is by design. While it was tempting to try to design IMP with a more organic voice, in part to try to make the human performer forget that IMP was not in fact human, I ultimately decided that giving IMP a voice that would *remind* the performer that IMP was not human was a better path. IMP sounds like a computer, because IMP is a computer. The purpose of IMP is not to emulate a human. The experience of improvising with a human would be much more efficiently achieved by playing music with a human improviser. IMP's synthetic sound serves as a reminder to both the performer and audience that this interaction may be similar to human interaction, but it is not the same. Keeping this distinction in mind is important, because the human's expectations are a large

part of the system that determines the success of a performance with IMP. The human performer's ability to deal with IMP on a purely musical level, and to avoid expectations of specific human-like behavior, is crucial to a musical outcome. Similarly, when two humans freely improvise together, specific expectations can inhibit the musical interaction. IMP's sound serves as a reminder to deal with IMP on the terms presented.

The synthesis module uses frequency modulation (FM) synthesis to generate its sounds.[47] Experiments were done with additive synthesis as well, but FM provided rich spectra and desirable timbral control in a less computationally expensive manner. FM synthesis involves modulating the frequency of one oscillator (the carrier oscillator) with the signal of a second oscillator (the modulating oscillator). The ratio of the frequencies of these two oscillators is the harmonicity ratio. The baseline setting for the harmonicity ratio in the primary FM pair is 2, meaning that when the frequency of the carrier oscillator is 100 Hz, the frequency of the modulating oscillator is 200 Hz. The timbral noise analysis module can change this ratio, but its basic setting is 2. The modulation depth is the amplitude of the modulating oscillator, and this controls the intensity of the modulation. The modulation depth on the primary FM pair is 1. This is the default setting, and it is not affected by the MPS. One could manually change this setting if desired.

IMP also employs a second order FM synthesis, meaning that there is a second modulating oscillator, which modulates the product of the first order oscillators. This second order oscillator has a harmonicity ratio with the carrier oscillator of 3.1467. This is an arbitrary

[47] For detailed explanations of the principles of FM synthesis, please see: John Chowning, "The Synthesis of Complex Audio Spectra by Means of Frequncey Modulation," in *Foundations of Computer Music*, ed. Curtis Roads and John Strawn(Cambridge, MA: The MIT Press, 1985). and Curtis Roads, "Modulation Synthesis," in *The Computer Music Tutorial*, ed. Curits Roads(Cambridge, MA: The MIT Press, 1996).

choice based personal taste, and it is not affected by the MPS. One could manually change this setting if desired.

Its modulation depth is controlled by a gain, which is controlled by the timbral noise analysis module. This means that at times the second order modulator is completely muted and does not affect the sound. This mechanism will be discussed in greater detail in the section on Machine Learning and Timbral Interaction.

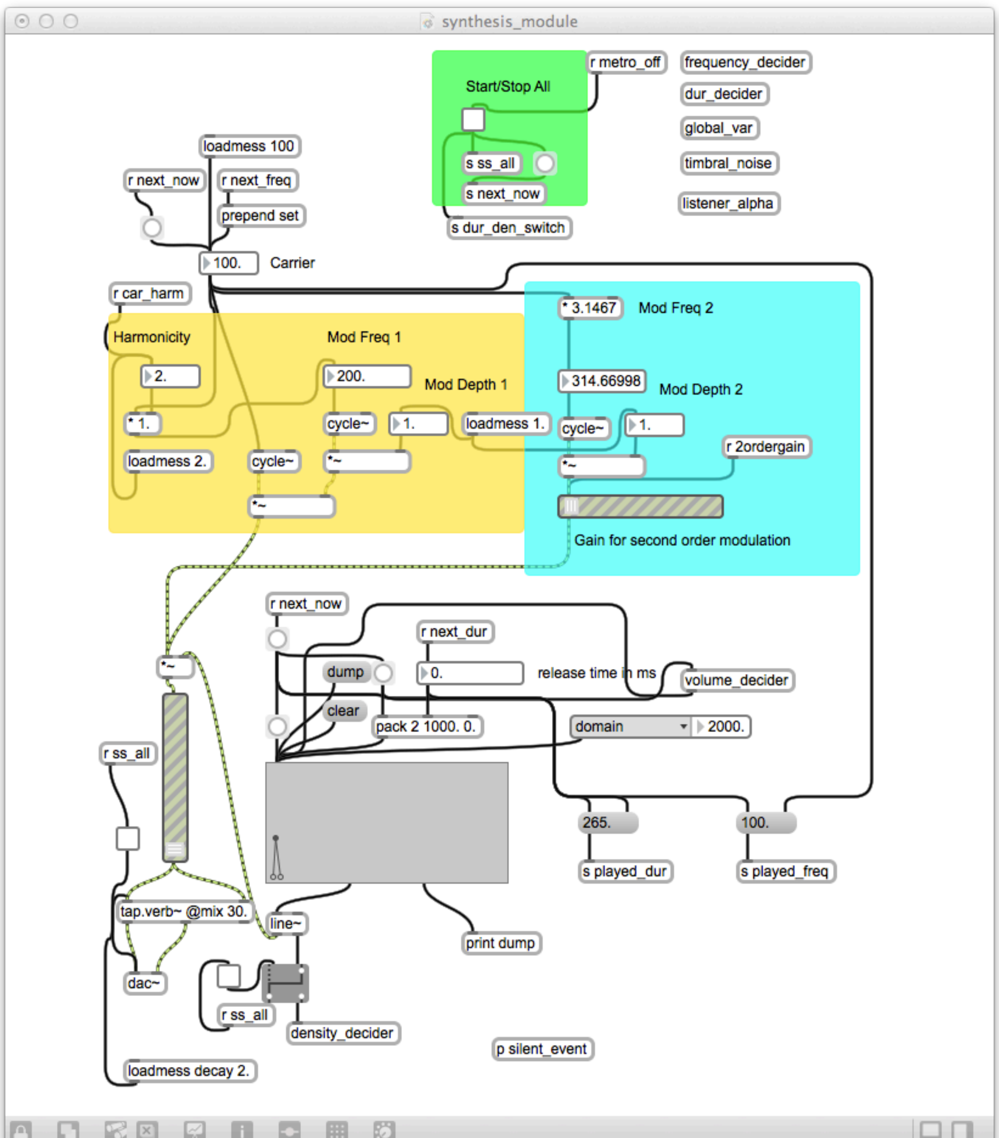

Figure 2. Synthesis Module

The product of the FM module is multiplied by a line function (a `line~` object) that creates an amplitude envelope for each event.[48] The envelope is a simple attack/release envelope, meaning that once the event reaches its maximum amplitude, the release stage begins and the amplitude steadily ramps down to zero. Once this envelope reaches zero, a bang is sent that triggers the next event.[49] This bang, which travels on the send/receive channel "`next_now`" is the cue to start all of the activity that results in the next event. The envelope flows into a master gain fader that is connected to a `tap.verb~` reverb object, and then out to the DAC. The default settings on the reverb are a mix of 30% and a decay of 2 seconds.

The synthesis module also contains three smaller decision modules: the `density_decider`, the `volume_decider`, and the sub-patcher called `silent_event`. The purpose of each of these modules will be described here, and the specifics of the decision process will be described later in 5.4 Decider Design.

The `density_decider` is the module that controls the density of events. Once the `line~` object (amplitude envelope) completes its process, it sends out a bang that enters the `density_decider`. A decision is made as to whether the next event will make sound, or be silent. This decision is weighted by the values in the density parameter of the MPS. The densest setting will have every event make sound, and the least dense setting will have no events make sound. If the next event is to have sound, the `density_decider` sends a bang on the "`next_now`" channel, which cues the synthesis process, and if the event is to be silent, a bang is

---

[48] I will use the term "event" to describe what might often be called a "note" in most musical discourse. This is in keeping with my desire to stay free from the note/rhythm abstractions, and it also allows for the existence of events that have no volume (rests) to be treated the same as events that do have volume, which is how they look to IMP.

[49] "The bang message has a specific use within Max – it's the message that tells many objects to *do that thing you do*." http://cycling74.com/docs/max6/dynamic/c74_docs.html#basicchapter02

sent to the silent_event sub-patcher. The silent_event sub-patcher receives the next duration from the duration decider and lets that amount of time pass in silence before sending a bang to the density_decider to cue the decision process for the next event. This system allows for the MPS to control the density of the texture without necessarily changing any of the other duration parameters, so it is possible for IMP to play sparsely in a setting that still has a relatively short duration of events, or densely in a setting that has relatively long event durations. This will be explained further in the discussion of the duration decider module, and the MPS.

The volume_decider is a simple module that assigns a value to the peak point in the line~ object that represents the highest amplitude in the amplitude envelope. The volume_decider decides between a randomized amplitude between 0.6 and 0.9, and the current volume of the human as heard by the listener module. The MPS parameter of duration listenerness weights this decision.

The synthesis module also contains the Start/Stop All toggle, which when activated, turns on the DAC and all of the various metros that are involved in triggering events, as well as sending the first bang on the "next_now" channel.

5.2 MUSICAL PERSONALITY SETTINGS AND GLOBAL VARIATION

One of the original goals of this research was to design a system with variable sets of behavioral characteristics, or musical personalities. This is implemented in IMP through the Musical Personality Settings (MPS), which are seven separate parameters that influence various aspects of IMP's behavior. The parameters are: Density of Events, Length of Events, Rhytmic Regularity, Frequency Listenerness, Duration Listenerness, Melodicness, and Variation.

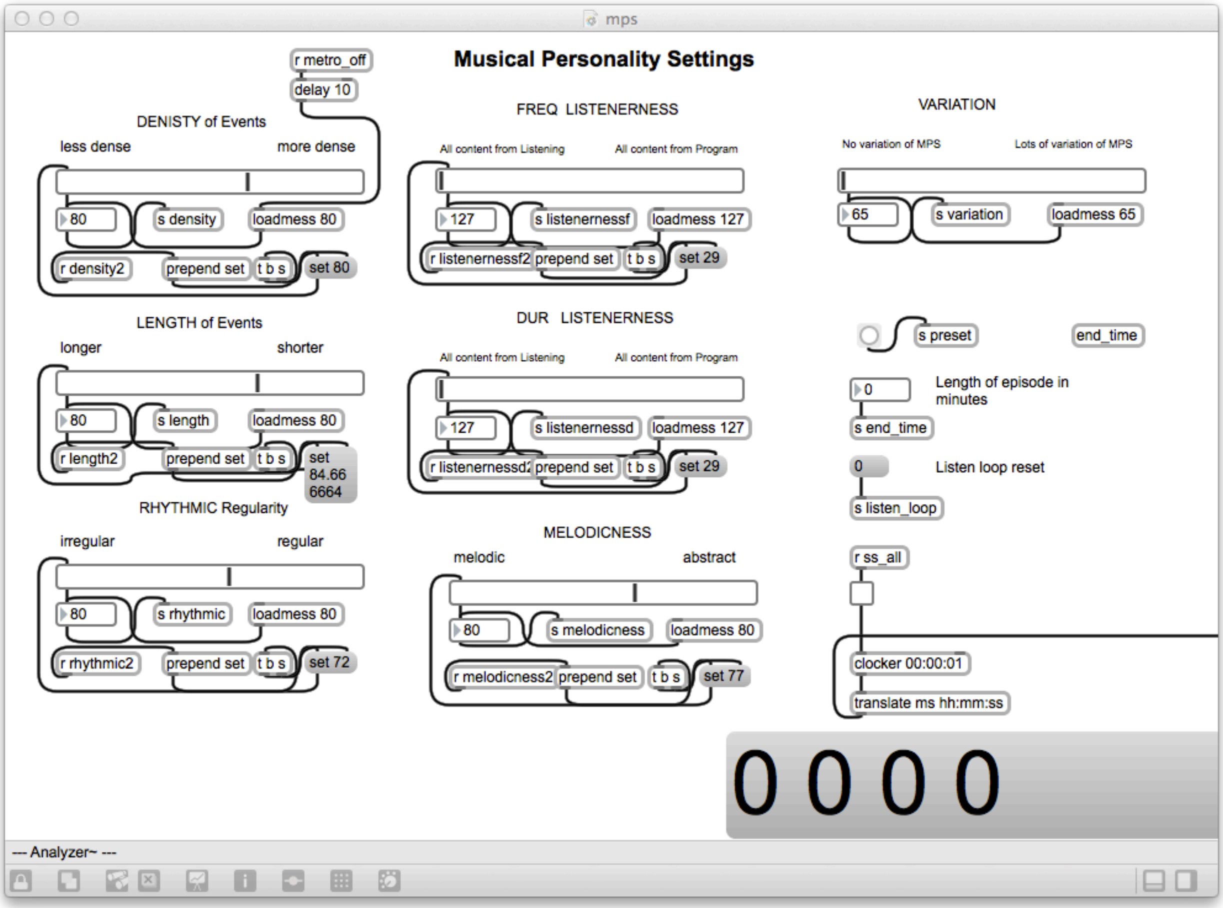

Figure 3. Musical Personality Settings

Each MPS parameter is controlled with a slider on the MPS interface. The interface also contains the mechanism for setting the length of an episode (or performance), and a visible timer to give the performer a reference for the amount of elapsed time since the beginning of the episode.

Ending an improvisation is a difficult task that requires great musical sensitivity when it is being negotiated amongst a group of humans, who can use visual signals and body language to assist in that tenuous and often vague communication. Negotiating a mutually agreeable end point with a computer is a monumental task, and one that has not yet been tackled with IMP. The MPS interface has a dialog that accepts a number for the “length of episode in minutes,” then

IMP will begin a process of stopping after that amount of time has passed. This process involves gradually lowering the density MPS until no more sounds are made, at which point the master start/stop mechanism is set to stop. While this method is not ideal, it does allow for the possibility of musical endings, even if it does not allow for a wide variety of textural possibilities in the last moments of an episode. The ending mechanism is an area that could be developed more fully in future versions.

The Density of Events parameter controls the weighting of the `density_decider`'s decision algorithm, which decides whether an event will make sound or not. The higher this parameter is set the higher the sound/silence ratio will be. This parameter is also influenced by what is heard from the human, once an episode begins.

The Length of Events and Rhythmic Regularity parameters work together to control IMP's tempo and sense of pulse. I use these terms (tempo and sense of pulse) loosely in this context, since there is no abstraction of meter present, but there can be a sense of IMP playing faster or slower, and in more or less regular event lengths. The Rhythmic Regularity parameter controls a pool from which duration proportions are chosen in the creative algorithm, and the Length of Events parameter controls a factor that controls the speed at which these proportions are realized. The specifics of this part of the system are discussed in 5.5 Creative Algorithm and Musical Memory Systems.

Listenerness is a term I have coined to describe the two parameters that control IMP's responsiveness to human input. The farther a listenerness slider is moved to the right, the more independently IMP will behave, and the farther left those sliders are moved the more IMP will derive its output from what it has heard from the human. There are two listenerness settings; one for frequency and one for duration. Frequency Listenerness controls the weighting of the

frequency decider mechanisms and influences whether IMP's pitch output is derived from its creative algorithm or from the pool of pitches it remembers hearing from the human. Duration Listenerness controls the weighting of the duration decider and similarly influences IMP's output in terms of duration of events.

The Melodicness parameter sets a set of pitches from which the creative algorithm chooses when IMP is generating content on its own. As the slider moves from left to right the pool of available pitches moves from pentatonic sets, through major scales, melodic minor (ascending) scales, diminished scales, whole tone scales, and finally to a fully chromatic set of pitches. Again, the specifics of this system are in 5.5 Creative Algorithm and Musical Memory Systems.

The final MPS parameter is Variation. This parameter weights the decisions made by the global variation module, which controls a mechanism that causes the variation of the other MPS parameters. A zero value in the Variation parameter would have no variations in the other MPS parameters (except as influenced by the human input), and the largest value would cause the MPS parameter variations to be most frequent and most extreme, although even at this maximum setting the variation of parameters would fall within a range that would be likely to maintain some musical continuity. The most often the parameters will change is once per second, and the longest interval between variations is 100 seconds, with the largest possible jump on any MPS scale being 10 units (on a 128 unit scale). This keeps IMP's output from seeming static in content, but helps avoid seemingly random huge shifts in musical space as well.

5.3 LISTENER SYSTEM

The listener module receives in the incoming audio signal from the human via the ADC, performs the audio feature extractions, and sends that extracted feature data to other IMP

modules or to The Wekinator. The central component in the listener module is the `analyzer~` object, which is a Max/MSP extension programmed by Tristan Jehan.[50] The `analyzer~` object outputs seven different audio features, and IMP uses five of those: pitch, loudness, brightness, noisiness, and attack.

Pitch is output as floating point MIDI note numbers and frequency in Hz. IMP uses the frequency in Hz, which is transmitted on the "`heard_freq`" channel and stored in the frequency decider module. The pitch estimation in `analyzer~` is based on Miller Puckette's `fiddle~` object.[51] The pitch estimation is the result of an analysis of the peaks in an FFT window, and application a likelihood function to give the most likely fundamental frequency.

Onset detection is done using a combination of pitch and amplitude analysis. A bang is sent out of the attack outlet whenever a new onset is detected. IMP is set to report a reattack if the amplitude changes more than 2 dB in a 100ms period of time. This setting is about 8 dB more sensitive than the default settings for `analyzer~`, but the testing has shown that recording too many events has a more desirable outcome than recording too few. This onset bang serves two important functions. The first is that it cues the current pitch to be sent to the "`heard_freq`" channel. The second is that it is sent into the `onset_average` subpatcher, which is used to keep a running average of the time between the last ten onsets detected from the human input, as well as to send the elapsed time between each individual onset on the "`heard_dur`" channel, which goes to the duration decider list of heard durations.

The `onset_average` subpatcher uses a `clocker` object to measure the elapsed time between bangs (which come from the attack outlet of `analyzer~` indicating a new onset). This

[50] Jehan.

[51] Miller S. Puckette, Theodore Apel, and David D. Zicarelli, "Real-Time Audio Analysis Tools for Pd and Msp," in *International Computer Music Conference* (Ann Arbor, MI: 1998).

time is the length of the last heard event. It is sent through a high-pass filter, and if it is longer than 50 ms, it is sent out as a heard duration. This high-pass filter was put in place because very short event durations were interfering with the envelope function in the synthesis module. When a total event duration that was shorter that the attack time of the envelope would be sent to the synthesis module, the `line~` object that creates the envelope would delete one of its points, and the sound would stop. This filter prevents that from happening and still allows for plenty of density of onsets, and the creation of short event durations.

The loudness, brightness, and noisiness features are sent via OpenSoundControl[52] to the Wekinator, where they are used to control the timbral interaction system, which will be described in 5.6 Machine Learning and Timbral Interaction. Each of these features is calculated from the FFT data. The loudness feature is based on spectral energy. The energy in each FFT bin is compared to en equal-loudness curve that approximates the frequency-based response of the human ear, and a single loudness value is output. This value is based on the combination of the amount of energy present and the location of that energy within the spectrum[53] The feature is reported on a dB scale (-96 dB to 0 dB) and this raw dB value is sent to the Wekinator, but the listener module also scales this value to a gain value between 0.0 and 1.0 for use in the `volume_decider` subpatcher of the synthesis module.

[52] Open Sound Control (OSC) is a protocol for communication among computers, sound synthesizers, and other multimedia devices that is optimized for modern networking technology. "Open Sound Control" http://opensoundcontrol.org/ (accessed March 19, 2013).

[53] Tristan Jehan and Bernd Schoner, "An Audio-Driven, Spectral Analysis-Based, Perceptual Synthesis Engine," in *Audio Engineering Society 110th Convention* (Amsterdam: 2001).

The brightness feature is simply the spectral centroid of each frame. The spectral centroid is the center of mass, or barycenter, of the spectrum.[54] The spectral centroid varies with the instrument as well as the brightness of a given tone within the range of sounds for that instrument. For example a very bright trombone sound will still have a lower spectral centroid than a dark flute sound. This is one of the reasons that it is suggested to use timbral training data that is instrument specific. This will be discussed further in 5.6 Machine Learning and Timbral Interaction.

The noisiness feature is calculated using the Spectral Flatness Measure (SFM), which is defined as, "the ratio of the geometric to the arithmetic mean of the energy per critical band."[55] This feature looks at the harmonicity of the overtones in the sound. Sounds with harmonic overtones, or overtones that are whole number multiples of the fundamental, are perceived as more pure. Sounds with non-harmonic overtones, or overtones that not whole number multiples of the fundamental, are perceived as being more noisy. The noisiness feature gives a single number on a scale from "pure" to "noisy." These features are used, along with loudness, by The Wekinator to drive the timbral interaction aspect of IMP.

5.4 DECIDER DESIGN

An early problem involved designing a mechanism that would allow IMP to make decisions. This basic decider mechanism is used in almost every instance in which IMP must

---

[54] Geoffrey Peeters, "A Large Set of Audio Features for Sound Description," (2004). http://recherche.ircam.fr/equipes/analyse-synthese/peeters/ARTICLES/Peeters_2003_cuidadoaudiofeatures.pdf (accessed February 24, 2013).

[55] Tristan Jehan and Bernd Schoner, "An Audio-Driven Perceptually Meaningful Timbre Synthesizer," *Proceedings of the International Computer Music Conference*, (2001).

choose between two options. MPS parameters, or other factors, are used to weight these random selections, so that a somewhat, but not completely, predictable behavior results.

An incoming bang starts the process by triggering a `tap.random` object to randomly select a number between -1.0 and 1.0. This random number is then scaled to fit the weighting parameter. Most of the weighting parameters have 128 possible values. This is the default resolution on the Max sliders, and this resolution is sufficient for the task of weighting these decision mechanisms. The slider is a simple interface for the human to use to set up the MPS, and these weights are all changed by other MPS and human musical input once the episode begins, so greater resolution of the sliders is not needed. After the random value is scaled, it is sent to an if/then object, and compared to the value from the weighting parameter, thus sending a bang out of one of two outlets. The outlet receiving the bang indicates the decision that has been made. It is a fairly simple mechanism that is employed throughout IMP.

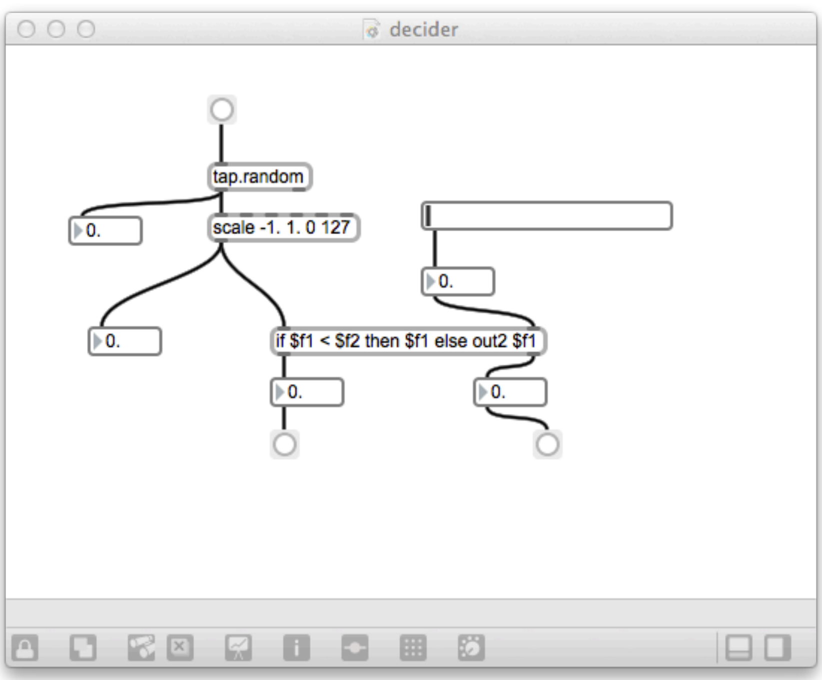

Figure 4. Basic decider

5.5 CREATIVE ALGORITHM AND MUSICAL MEMORY SYSTEMS

Equal and interactive participation in an improvised duo musical situation requires that each participant has the ability to both hear and remember what the other has played, the ability the create and introduce new musical content, and the option to choose between the two. IMP's creative algorithm and musical memory systems are what give IMP these abilities. There are two primary modules in the creative algorithm: the frequency decider, and the duration decider. Each functions very similarly, but their processes do not affect each other. The frequency decisions are made independently of the duration decisions, and vice versa.

The decision to keep these processes separate was mostly intuitive. In my own improvising practice I often like to deal with these musical components independently. For example, in a trio setting, I may use the pitch content from one collaborator, and apply the rhythmic content from the other to that pitch content. This structure, that treats these components independently, allows the possibility of IMP behaving similarly. Meaning that the human's rhythmic choices may have a greater affect on IMP than the human's frequency choices, or vice versa, and the MPS allow for different levels of listenerness in regard to each of these parameters.

The beginning of the new event process works similarly on both the frequency and duration deciders. A bang on the "`next_now`" channel cues each new event. That bang causes a decision to be made as to whether the next event will come from IMP's generative algorithm or IMP's pool of heard events. This is the way in which IMP can play off of what it has heard, or introduce independent new material. The frequency and duration deciders each have two sides: one side is a list of heard data, and the other side is the generative algorithm, or the part that makes IMP's original input.

On the heard data side of the duration decider, each new heard duration is entered into a list that keeps the last ten heard durations. When the "`next_now`" bang is sent to the heard data side, one of these ten most recently heard durations is selected randomly, and output as the next duration on the "`next_dur`" channel, which is received in the synthesis module and stored in the amplitude envelope until the `next_now` triggers an event and a new duration is sent. This side of the decider also keeps a running average of the list of heard durations that can be used to change the length and density MPS. This average is scaled to fit the MPS values, and every 500 ms a decision is made to change or not change the MPS based on the current average of heard durations. The average is scaled differently for the length and duration MPS, and the decision to change each MPS is made independently. This system keeps IMP in a similar density and speed area as the human improviser, but does allows for some divergence in terms of these parameters as well. It has the effect of varying how much it seems like IMP is following or ignoring the human.

On the generative algorithm side of the duration decider, IMP chooses a duration proportion from a set of lists of proportions, and that proportion is multiplied by a length factor to get the next duration. There are 15 different files of proportion values numbered 00 – 14, and one of these is loaded into the duration `coll` object (`dur_coll`) based on the rhythmic regularity MPS. The lower the rhythmic regularity MPS value the more varied the proportions are. The 00 file has 24 different proportions, and the 14 file has one proportion. The more similar duration proportions there are, the more of a sense of pulse one hears from IMP. There is a rhythmic variation decider that uses the variation MPS to change the choice of proportion `coll` file during the course of an episode. If the `next_now` bang is sent to the generative algorithm side of the duration decider, a proportion is output from the currently loaded duration `coll`.

Once a proportion is sent from the duration `coll`, it is multiplied by a length factor. This length factor is controlled by the length MPS, and ranges from 5.0 to 20.0. Since the proportions range from 10 to 125, the shortest duration that IMP will create from its generative algorithm is 50 ms, and the longest is 2500 ms (2.5 seconds). This proportion/length factor system allows IMP to deal separately with the sense of pulse and the sense of speed. Rhythmic regularity with long lengths will feel slow but have pulse and little rhythmic regularity and short lengths will feel fast, but with little feeling of pulse. The length factor is also influenced by the input from the human, so IMP will follow the human's tempo, for the most part, although as was mentioned earlier there is a mechanism in place to keep that following from happening too closely.

The frequency decider has a very similar structure to the duration decider. On the heard data side there is a list of the last ten heard frequencies, and each new heard frequency is added to that list. If a `next_now` bang is routed to the heard data side of the frequency decider, a frequency from the list of the ten most recently heard frequencies is selected. This randomly selected frequency is output as the `next_freq`, and a loop is setup that will output the rest of the list as the next frequencies. For example if the initial `next_now` bang causes the heard frequency in index 7 on the list to be chosen, then the next three frequencies sent will be indexes 8, 9, and 10. After the end of the list is reached, the system resets to choose the next frequency from either the heard data side or the generative algorithm side. This loop system causes IMP to play not just one pitch that it has heard from the human, but a series of pitches, and in the same order that they were heard.

The generative algorithm side of the frequency decider is structured similarly to the generative side of the duration decider. There are 42 different files of sets of pitches, and one of those files is loaded into the frequency `coll` (`freq_coll`) based in the melodicness MPS. The

files are numbered 00 – 41. The lower numbered sets are major pentatonic scales, and as the numbers go up they cycle through major scales, ascending melodic minor scales, diminished scales, whole tone scales, and finally a chromatic scale. There is a `melody_decider` that changes the choice of frequency `coll` file, according to the variation MPS, during the course of an episode.

Once a frequency is sent out of the current frequency `coll`, a loop is enabled that will select the next 1-5 pitches in a stepwise relationship to the original pitch within the frequency `coll`. The steps may move up or down, or any combination of up and down. This feature gives IMP's output a little more melodic coherence. While it does not eliminate large melodic leaps, it does force at least occasional stepwise motion. Each frequency is sent out on the "`next_freq`" channel, which is received in the synthesis module and stored as the frequency of the carrier oscillator until a `next_now` bang triggers an event and a new frequency is generated.

It should be noted that these frequency `coll`s are the one place in IMP where pitch information is stored using MIDI note numbers. Once the MIDI note number is sent out of the `coll`, it passes through a MIDI to frequency conversion, and is handled as a raw frequency from that point on. The choice was made to use MIDI note numbers in these files to keep the creation of those files simple, and to keep them somewhat human readable.

5.6 MACHINE LEARNING AND TIMBRAL INTERACTION

Much of the communication between participants in non-idiomatic improvisations is timbral. Timbre is equal to pitch and rhythm in many free improvisation encounters, so it is important that IMP has both a way to hear timbre, and a way to communicate timbre. The mechanism for hearing timbre was described in 5.3 Listener System, and the mechanism for communicating timbre was described partially in 5.1 Synthesis Module. This section will

describe the way the hearing mechanism is connected to the synthesis module and the logic behind those connections.

The Wekinator is a real-time machine learning application by Rebecca Fiebrink, and it is based on Weka, which is an earlier machine learning application that does not function in real-time. While IMP is playing, the Wekinator is running as a separate application on the same computer. IMP's listener module sends loudness, brightness, and noisiness data to the Wekinator via OSC. The Wekinator runs these three streams of data through a neural network that outputs a single value between 0 and 127, which is sent back to IMP via OSC where it controls the timbral elements of the synthesis module.[56]

The Wekinator must first be trained by playing tones into the feature extractor (which is part of the listener module), and assigning a value between 0 and 127 to each sound played in. This is usually done with 0 being the most pure tone, and 127 being the noisiest tone. However, if one wanted IMP to respond differently in the timbral domain, one could train the Wekinator differently. When IMP gets a 0 from the Wekinator, IMP plays its most pure tone, and a 127 gives its noisiest tone, with the varying degrees in between. With that knowledge, the Wekinator could be trained for any given input to make pure tones or noisy tones, as long as that input is associated with that value in the training stage.

For most of IMP's testing I used a set of training data comprised of solo trombone and solo saxophone recordings. This was done in hope that one universally useful set of training data could be used for all performers with IMP. That may still be possible, but a much larger sample size will be needed, so individual instrument training sets have been devised which have proved

[56] A detailed explanation of the neural network algorithm can be found in: Rebecca Fiebrink, "Real-Time Human Interaction with Supervised Learning Algorithms for Music Composition and Performance" (Dissertation, Princeton University, 2011). http://www.cs.princeton.edu/~fiebrink/Rebecca_Fiebrink/thesis.html

to be more accurate with smaller amounts of training data. When a trombonist performs with IMP, the Wekinator is trained on trombone sounds, and when a flutist performs, the training data is all flute sounds, etc.

The value returned by the Wekinator is received in the timbral noise module. This incoming value is in constant flux, so the timbral noise module polls that value every 50 ms and keeps a running average of the ten most recent polled values, and this average is what is used to drive the timbral variations in IMP's sound. Using this running average smooths the data flow, creating a more organic, less scattered result.

The value from the Wekinator is tied to the gain on the second order modulation oscillator in the synthesis module. This means that when the human is playing pure tones, the second order modulation is turned off. As the human's sounds get noisier, the second order modulation depth is increased and IMP's tone gets more strident. After a certain threshold, the harmonicity ratio on the first order modulation begins to change to a non-harmonic ratio as well, which can get quite crunchy. This direct relationship between the timbre of the human input and the timbre of IMP is the way I prefer to play with IMP, but it is entirely dependent on how the Wekinator is trained. Different training data can produce very different results.

CHAPTER 6: USER'S GUIDE

IMP is built on a commercially available platform, and it should be relatively easy for anyone with the proper software to download the IMP files and perform or play with IMP. The majority of the IMP files are Max 6 files, with the exception of the frequency and duration `coll` files, which are simple .txt files. To run IMP, one must have Max 6, the Wekinator, the `analyzer~` external, and Tap Tools, which is a set of objects and extensions for Max 6.[57] Max 6 and the Wekinator are available for Mac OS X and Windows. The newest version of Tap Tools is OS X only, but there are older versions for Windows. It is likely that IMP would work in a Max/MSP 5 environment using Tap Tools 3 on OS X or Windows, but this has not been tested. All of the testing has been done with Max 6 and Tap Tools 3.6.4 on Mac OS 10.8.2.

A few simple steps must be taken to get IMP ready to run on a new machine. The IMP package contains two directories ("`dur_colls`" and "`freq_colls`"), which hold all of the .txt files that get loaded into the duration decider and frequency decider and used in the generative algorithm. The path to these directories must be specified in the `freq_decider` and `dur_decider` patches. Max 6 is capable of using relative paths in relation to the location of the application, but since each user may put these directories in different places relative to the application, the easiest way is simply to specify the absolute path. In the `dur_decider` specify the absolute path to `dur_colls` and in the `freq_decider`, specify the absolute path to `freq_colls`. See figure 4 for a detailed screenshot of the area where the path should be entered.

[57] Max 6 is a commercial program and is available from http://cycling74.com/products/max/. Analyzer~ external, Tap Tools and the Wekinator are free downloads and are available from http://web.media.mit.edu/~tristan/maxmsp.html, http://74objects.com/taptools/ and http://wekinator.cs.princeton.edu/ respectively.

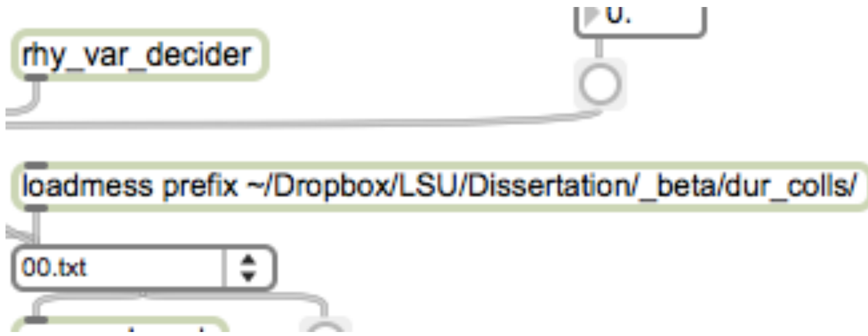

Figure 5. Section of `dur-decider` where absolute path to `dur_colls` must be specified.

Once the paths to the `coll` directories are set, open the Wekinator. The first tab is "Chuck & OSC Setup." Select "Edit Chuck configuration" and on the "Synthesis (output)" tab select "Use a different Max/OSC synth module." Click ok. Click the "Run" button to start ChucK. The "Features Setup" tab will appear. Check the box next to "Custom OSC feature extractor" and put a 3 in the box next to the word features. Then click "Go!" at the bottom of that tab. The "Learning Setup" tab will appear. The IMP package has a number of .arff files, which are dataset files that can be read by the Wekinator. If you wish to use one of these files, select "choose file" in the "Configure dataset" pane. If you wish to provide your own training data, select "create new dataset." Click "Go!" and the bottom of the tab, and the "Use it!" tab should appear. At this point you can collect (more) data or train the dataset that is loaded (or collected). After the dataset is trained, the "Run" view should be available, and once you click "Run" the Wekinator will receive features from the listener module and output timbral noise data to the timbral noise module.

IMP can now be run with only two windows visible. All of the controls and settings can be accessed from the `synthesis_module.maxpat` and `mps.maxpat` files. In the MPS window, each MPS parameter slider should be set as desired. On the right side of the window, there is a preset button that should be clicked. This prefills the heard data lists with generic

values so that IMP has “heard” data from which to play even before the human has played ten events. Without this preset button, IMP would remember the heard data from its last partner, which at times may be awkward. Below the preset button is a number box marked “Length of episode in minutes.” The number put in this box will determine how much time will elapse between the beginning of the episode and when IMP’s ending process is started. This number should be greater than or equal to 1.

In the synthesis module window, there are two main controls that need attention from the performer. The first is the gain slider on the lower left of the window. This is the master volume for IMP’s output. It should be set to about 75%, but can be adjusted as needed. If you start IMP and do not hear anything, this is likely the thing you forgot to do. There is a toggle marked “Start/Stop All” in a green box at the top center of the synthesis module and does exactly what it says. This toggle turns on the ADC/DAC, as well as all of the various metros that are used by IMP, and switches gates that control data traffic as well. Turning this toggle off will turn off the DAC and stop the other processes. Next to the green box is a delayed start button. Clicking that button will give a one second pause, then turn on the “Start/Stop All” toggle. This feature is handy when the performer is also turning IMP on, as it gives one time to be prepared to play at the same time IMP starts. Adjusting the number in the delay box will shorten or extend the length of the pause. IMP will now play until the amount of time set in the “Length of episode” box has elapsed, at which point IMP will start its ending process. If you wish to stop IMP before that amount of time has elapsed, just click the “Start/Stop All” toggle.

CHAPTER 7: ASSESSMENT OF SUCCESS

Success in this research is defined as a musically viable experience for both the performer and the audience. Since dialogic communication between human and IMP is a foundational concept of this research, determining how well that communication occurs is also a priority. Most of the early testing was done with me improvising on trombone with IMP, so as a means to determine the success of IMP, a number of other noted improvisers have performed with IMP, and completed a survey about the experience. These complete surveys are reproduced in Appendix 3.

On January 31, 2013, five New Orleans area improvisers played 3-8 minute episodes with IMP in a studio setting. I was the only other observer. These musicians were: Rick Trolsen, Mark McGrain, Janna Saslaw, Ray Moore, and Brad Walker. This was also the first time I was able to hear IMP interact with other people, causing me to make some changes to the program. Most were small matters of housekeeping, but one was a fairly significant structural change.

The reactions of the individual musicians were somewhat varied. One musician did not like it at all, two of the musicians responded quite positively, and two were more in the middle of that range. These responses were shaped as much by the musician's own expectations, preconceptions, and attitude towards the program as they were by the behavior of IMP, but it is difficult to control for the human mind in an experiment. It should also be noted that this is a fairly small sample, so while this process had great value in terms of letting me hear musicians with whom I am familiar play with IMP and give me feedback, it do not believe that there is any statistical significance to the responses.

The survey consisted of five Likert Scale questions and a space for additional comments. The statements that the respondents were asked to rate on a scale of 1 = strongly disagree to 5 =

strongly agree were: What I played influenced what IMP played, What IMP played influenced what I played, There was musical communication between IMP and myself, Playing with IMP was a musical experience, and I would like to hear others play with IMP.

Trombonist Rick Trolsen was the first to play on that day. His only comment was that he would be interested in more development of the program. He did not feel that there was musical communication between himself and IMP, but he did feel that playing with IMP was a musical experience and he indicated that he would like to hear others play with IMP.

The next participant was also a trombonist. Mark McGrain left no written comments, but did spent quite a bit of time talking with me after he played. He was very interested in the ideas, and seemed to have his own strong opinions about how a system like IMP should behave. He had a better assessment of the presence of musical communication than Rick did, but a lower appraisal of the musicality of the experience. Both Rick and Mark gave a 2 (somewhat disagree) to the statement that what they played influenced what IMP played, and a 3 to the statement that what IMP played influenced what they played.

Janna Saslaw is a flutist and music theory professor, and her experience with IMP was quite different. She strongly agreed (marked a 5) with the statement that what she played influenced what IMP, and the statement that what IMP played influenced what she played, yet she rated a 3 for the statement that there was musical communication between her and IMP. She remarked that she felt that IMP followed her too closely, at times playing the same note that she played, which she did not find to be a rewarding practice. In hindsight, we realized that this was due to the fact that I had moved both followerness MPS settings to their highest level. This was done in the previous session with Rick Trolsen, because Rick was asking about the settings, so we did a number of episodes with different settings.

Saxophonist Ray Moore gave the highest numbered survey responses. He rated each statement a 5, except for the 4 that he wrote when ranking if what he played influenced IMP. He did comment that the interaction seemed to have some latency and that he wished that the interaction was more "real-time," but he definitely indicated that communication was happening. It should be noted that Ray is a close friend of mine and even though I asked him to be as critical of the program as he liked, and told him that negative responses were just as helpful as positive ones, it is possible that our personal relationship influenced his responses. It is also possible that Ray just likes playing music, even with a computer.

The last musician to play with IMP that day was Brad Walker, who played tenor saxophone. Brad gave the lowest number marks of any of the participants. He did indicate that IMP seemed to be responding to what he played, but he called it, "superficial at best." Listening to his session, I tended to agree with his assessment that it was not a musical experience. Brad and IMP did not find any sympatico on any level.

This afternoon of hearing others play with IMP caused me to make a number of changes to the program. Some were basic, such as putting the high pass filter on the heard duration mechanism and making a preset button to clear IMP's memory of one player before another plays. These were simply revisions to fix quirks of the program that revealed themselves with multiple episodes with different players. However, there were also two much more significant changes that were a result of this day of testing.

As I had my first chance to listen carefully to others play with IMP, I realized that I was disappointed with the way IMP was playing in a melodic sense. The intervals seemed too wide and a bit unnatural. Keeping in mind George Lewis's advice that it should make music I like, I added the loop functions on each side of the frequency decider. These loops encourage more

linear intervals for at least short parts of a series of notes (melody). This adjustment both helped the musical flow of the lines made by IMP, but I think it also helped the humans who play with IMP to connect with the material that IMP s bringing to the interaction.

The other big observation that came from this first day of testing was that while IMP did seem to respond to the human's input, there was a significant latency in that response. This caused a major change in the structure of the creative algorithm. The original design had three sections in each decider (frequency and duration). The heard data section, the generative algorithm section, and a middle section that kept track of the last 40 events that were either played by IMP or heard from the human. The thought behind this was that it would enable the human and IMP to have a shared space of material. What it actually did was just make IMP slower to play any of the material it heard from the human. The removal of this middle pool, and the addition of the "melodic" loops in the frequency decider helped make IMP's response time to the human much quicker, and gave IMP's output a better flow.

To test these changes, another round of play testing was held on February 19, 2013. Mark McGrain, Janna Saslaw, and Brad Walker participated in this second round. Each participant gave more agreeable responses to the survey after the February 19 session than they did after the January 31 session. Janna Saslaw commented that she thought that IMP, "responded more 'artistically' to [her] input." Brad Walker noted that IMP seemed to be more "pliable and responsive," but still did not find the experience to be particularly musically enjoyable.

It is likely that IMP simply does not inhabit an aesthetic space that Brad enjoys. This is encouraging, in a way, since it can be seen as confirming that IMP does function within a certain aesthetic framework; even if it is one that is not shared by one of the testers. This idea is

confirmed by the enthusiasm shown for the experience by Janna and Ray, who have different aesthetic leanings than Brad.

The second testing session was also the first time that I heard someone else play with IMP and had a completely positive reaction to the music I was hearing. In both Janna and Mark's sessions there were episodes in which I sat and listened and felt like I was having a musical experience as an audience member. The interaction, cooperation, and moments of surprise that I value in improvised music performances were all present in the sessions with Mark and Janna. As a listener, I also enjoyed moments of Brad's session, likely more than Brad did.

On Friday, March 1, 2013, I performed with IMP as part of the opening concert of the Inaugural Integrated Composition Improvisation and Technology Symposium at the University of California – Irvine. IMP had been tested extensively in my personal studio, and I had heard others play with IMP in private controlled settings, but this was the first public performance with IMP.

I realized in testing that much of the musical success of an interaction with IMP depends upon the human's ability to maintain musical intentions, and not try to control or test the program. When I (or one of the testers) started to think about how IMP was reacting to my input, instead of just trying to react musically to IMP's actions, and trust IMP to react musically to mine, the episode was usually musically unsuccessful. The idea that dealing solely with musical intentions and avoiding over-thinking on any subject is a good space for improvised music is not unique to interactions with machines. It is a good practice for interactions with other humans as well.

During the performance in Irvine, I was able to keep focused on the music, and I had an enjoyable experience. The music felt good to me while it was happening, and I felt like IMP and

I were interacting in musical ways. I received a number of comments from audience members after the concert and they were all positive, which could be an example of the manners of the audience members more than the success of the performance. Several of the other symposium participants asked specific questions about the design of IMP and the reasons behind those design decisions. These conversations indicated that the performance, and IMP itself, was interesting. I was not able to collect any sort of survey data after the performance, but anecdotal evidence and my own feelings about the performance lead me to believe that it was a success.

Having spent significant amounts of time playing with IMP myself, and hearing other musicians whom I respect play with IMP, I consider the research to be largely successful. There is dialogic communication between the humans and IMP. The results are musically enjoyable, at least within a particular aesthetic. The MPS do allow for the shaping of IMP's tendencies in ways that are consistent but not completely predictable. Reaction time is an area that thus far has been less successful. There is latency between the human's action and IMP's adoption of that action (when IMP chooses to adopt that action). This is counter to what we expect when improvising with other humans, which is usually immediate adoption or immediate rejection of the incoming musical ideas. This causes the human to develop new strategies for improvising with IMP, which can be frustrating for the human participants, but it is also sometimes quite rewarding, since the development of new strategies is often a valued undertaking amongst improvisers.

CHAPTER 8: CONCLUSION

As was stated in Chapter 5, my goal in the creation of this Interactive Musical Partner (IMP) software was to make a software system that could live up to all three parts of its name. It would have to be interactive, musical, and a partner. IMP has met these basic goals, and done so in a manner that is relatively simple. There is complexity in the sounds that IMP makes, but the algorithm that generates those sounds is not overly complex.

My intention in the creation of this system was not to model the process on the way the human brain works while improvising, but to create a more simple system that could generate similar results. Improvising with IMP is not exactly like improvising with another human, but it is an aesthetically valuable experience. Playing with IMP requires a different mindset than playing with anther human, just as certain human partners require different mindsets than others. IMP has a musical personality, and even though its behavior can be shaped through the MPS, there is an underlying IMP-ness that must be dealt with regardless of the way the MPS is set.

Of course there is much room for further research. Application of a melodic segmentation algorithm to both the listening and generative processes could be useful, although making every melody specifically strong would inhibit the craziness that IMP can display in its current state, and that craziness can be desirable. A set of separate genre models might also prove useful. A system where IMP could shift to a generative model that would follow generic rules when it sensed that the human was displaying the characteristics of a genre. This would require the construction of generative algorithms that could play in a certain genre, and then devising a system to move between those genres in a musical fashion. This would make it easier for IMP to play with a wider range of musicians; particularly with musicians who have a certain “bag” that

they are unwilling to leave. It would also allow IMP to leave its "bag" from time to time, or maybe expand its "bag" would be a more accurate way to put it.

I learned much during this research. Some of what I learned was about programming generative algorithms, and machine learning, and FM synthesis, but most of what I learned was about me and my musical tastes and preferences and desires. I learned that what I enjoy in listening to improvised duets is the interaction. I enjoy hearing the ways that one musician will musically accept or reject the ideas and suggestions of the other. I learned that brash displays of great technique get less of a reaction from me than a well-placed surprising sonority.

I learned that a significant portion of the communication that happens between humans when freely improvising is extra-musical. A head nod, a peek from one eye, or a tilt of the instrument can go a long way towards showing agreement that an ending point has been reached, or giving confirmation that a musical message has been received. The future research that will really take IMP, and similar pieces of software, to the next level is the research that will devise a way for that extra-musical communication to happen.

Designing a program intended to play with an improviser, and then improvising with that program has caused me to assess every aspect of my practice as an improviser. Not only has this research expanded my understanding of the technologies involved and made me a better technologist, but striving to get the technology to be musical has made me look at all sides of the music I make, and it has made me a better improvising artist.

BIBLIOGRAPHY

"Open Sound Control" http://opensoundcontrol.org/ (accessed March 19, 2013).

"Virtual Composer Creates New Music" http://www.npr.org/templates/story/story.php?storyId=113719483 (accessed May 22, 2012).

Appleton, Jon H. and Don Cherry. *Human Music*. New York: Flying Dutchman, 1970.

Bailey, Derek. *Improvisation: Its Nature and Practice in Music*. New York: Da Capo Press, 1993.

Belgrad, Daniel. *The Culture of Spontaneity: Improvisation and the Arts in Postwar America*. Chicago and London: The University of Chicago Press, 1998.

Benson, Bruce Ellis. *The Improvisation of Musical Dialogue: A Phenomenology of Music*. New York: Cambridge University Press, 2003.

Borgo, David. *Sync or Swarm: Improvising Music in a Complex Age*. New York: Continuum International Publishing Group, 2005.

Braxton, Anthony and Richard Teitelbaum. *Silence / Time Zones*. [S.l.]: Black Lion, 1996.

Chadabe, Joel. *Electric Sound: The Past and Promise of Electronic Music*. Upper Saddle River, NJ: Prentice Hall, 1997.

Chowning, John. "The Synthesis of Complex Audio Spectra by Means of Frequncey Modulation." In *Foundations of Computer Music*, edited by Curtis Roads and John Strawn, 6 - 29. Cambridge, MA: The MIT Press, 1985.

Douglas, Robert L. "Formalizing an African-American Aesthetic." *New Art Examiner* 18, (1991): 18-24.

Evans, Bill, Harry Evans and Steve Allen. *The Universal Mind of Bill Evans: The Creative Mind and Self-Teaching*. New York, N.Y.: Rhapsody Films, 2005.

Fiebrink, Rebecca. "Real-Time Human Interaction with Supervised Learning Algorithms for Music Composition and Performance." Dissertation, Princeton University, 2011.

Fiebrink, Rebecca, Dan Trueman and Perry Cook, "The Wekinator: Software for Using Machine Learning to Build Real-Time Interactive Systems " http://wekinator.cs.princeton.edu/ (accessed November 29, 2011).

Harley, James. "Computational Approaches to Composition of Notated Instrumental Music: Xenakis and the Other Pioneers." In *The Oxford Handbook of Computer Music*, edited by Roger T. Dean. Oxford: Oxford University Press, 2009.

IRCAM, "Improtech Paris/Ny 2012: Improvisation & Technology" http://repmus.ircam.fr/improtechpny (accessed May 27, 2012).

IRCAM, "The Omax Project Page" http://omax.ircam.fr/ (accessed May 27, 2012).

Jehan, Tristan, "Max/Msp Externals" http://web.media.mit.edu/~tristan/ (accessed January 27, 2013).

Jehan, Tristan and Bernd Schoner. "An Audio-Driven Perceptually Meaningful Timbre Synthesizer." *Proceedings of the International Computer Music Conference*, (2001).

Jehan, Tristan and Bernd Schoner. "An Audio-Driven, Spectral Analysis-Based, Perceptual Synthesis Engine." In *Audio Engineering Society 110th Convention*. Amsterdam, 2001.

Johnson, George. "Undiscovered Bach? No, a Computer Wrote It." *The New York Times*November 11, 1997.

Lewis, George. "Mobilitas Animi: Improvising Technologies, Intending Chance." *Parallax* 13, no. 4 (2007): 108-122.

Lewis, George E. "Purposive Patterning: Jeff Dondaldson, Muhal Richard Abrams, and the Multidominance of Consciousness." *Lenox Avenue: A Journal of Interarts Inquiry* 5, (1999): 63-69.

Lewis, George E. "Too Many Notes: Computers, Complexity and Culture in "Voyager"." *Leonardo Music Journal* 10, (2000): 33-39.

Lewis, George E. "Living with Creative Machines: An Improviser Reflects." In *Afrogeeks: Beyond the Digital Divide*, edited by Anna Everett and Amber J. Wallace, 83-99. Santa Barbara, CA: Center for Black Studies Research, University of California, 2007.

Lewis, George E. *A Power Stronger Than Itself: The Aacm and American Experimental Music*. Chicago and London: The University of Chicago Press, 2008.

Peeters, Geoffrey. "A Large Set of Audio Features for Sound Description." (2004). http://recherche.ircam.fr/equipes/analyse-synthese/peeters/ARTICLES/Peeters_2003_cuidadoaudiofeatures.pdf [accessed February 24, 2013].

Place, Tim and Jesse Allison, "Tap Tools" http://74objects.com/taptools/ (accessed March 19, 2013).

Puckette, Miller S., Theodore Apel and David D. Zicarelli. "Real-Time Audio Analysis Tools for Pd and Msp." In *International Computer Music Conference*. Ann Arbor, MI, 1998.

Roads, Curtis. "Modulation Synthesis." In *The Computer Music Tutorial*, edited by Curits Roads, 213 - 262. Cambridge, MA: The MIT Press, 1996.

Rowe, Robert. "Personal Effects: Weaning Interactive Systems from Midi." In *Proceedings of the Spark Festival 2005*. Minneapolis, MN, 2005.

Solis, Gabriel and Bruno Nettl, eds. *Musical Improvisation: Art, Education, and Society*. Urbana and Chicago: University of Illinois Press, 2009.

Spear, Andrew D., "Husserl on Intentionality and Intentional Content", Internet Encyclopedia of Philosophy: A Peer-Reviewed Academic Resource http://www.iep.utm.edu/huss-int/ - SH1a (accessed February 6, 2013).

Teitelbaum, Richard. "Improvisation, Computers, and the Unconscious Mind." *Contemporary Music Review* 25, no. 5/6 (2006): 497-508.

Teitelbaum, Richard, Anthony Braxton and George Lewis. *Concerto Grosso for Human Concertino and Robotic Ripieno*. Therwil, Switzerland: Hat Hut Records, 1988.

APPENDIX 1: BASIS RECORDINGS

Abrams, Muhal Richard, Fred Anderson and George Lewis. *Sounddance Duos with Fred Anderson and George Lewis*. New York, NY: Pi Recordings, 2011.

Albert, Jeff, Dave Rempis and Jeb Bishop. 2008. "Duos (Chicago) 2008," jeffalbert.com, http://jeffalbert.com/?p=145.

Appleton, Jon H. and Don Cherry. *Human Music*. New York: Flying Dutchman, 1970.

Barry, Robert and Fred Anderson. *Duets 2001*. Chicago, IL: Thrill Jockey Box, 2001.

Bishop, Jeb, Josh Abrams, Hamid Drake, Mats Gustafsson, Fredrick Lonberg-Holm, Wadada Leo Smith and Ken Vandermark. *98 Duets*. Chapel Hill, N.C.: Wobbly Rail, 1998.

Braxton, Anthony, Taylor Ho Bynum and Dram, "Duets (Wesleyan) 2002", DRAM http://www.dramonline.org/Identifier/IN576.

Braxton, Anthony, Taylor Ho Bynum and Services Naxos Digital. *Braxton, A. Compositions 304 and 305 / Bynum, T.H.: Scrabble / to Wait / All Roads Lead to Middletown (Duets, Wesleyan, 2002)(Braxton, Bynum)*. Hong Kong: Naxos Digital Services Ltd., 2010.

Braxton, Anthony and Joe Fonda. *Duets 1995*. S.l.: Clean Feed - Trem Azul, 2007.

Braxton, Anthony and Gerry Hemingway. *Old Dogs. (2007)*. New York, NY: Mode Avant, 2010.

Braxton, Anthony, Dave Holland, Leo Smith and Richard Teitelbaum. *Trio and Duet*. TORONTO: SACKVILLE, 1974.

Braxton, Anthony and George Lewis. *Elements of Surprise*. Moers: Moers Music, 1976.

Braxton, Anthony and Richard Teitelbaum. *Silence / Time Zones*. [S.l.]: Black Lion, 1996.

Brøtzmann/Bennink. *Total Music Meeting 1977 Berlin*. CD. Bro B, 2006.

Lèandre, Joëlle and George Lewis. *Transatlantic Visions*. [Paris]: Rogue art, 2009.

Lewis, George E. and Roscoe Mitchell. *Voyager Improvised Duos between Human and Computer Musicians*. Tokyo: Avant, 1993.

Lewis, George and Douglas Ewart. *Jila - Save! Mon. - the Imaginary Suite*. Milan: Black Saint, 1979.

Lewis, George, Steven Schick, Quincy Troupe, Sarah Cahill and Orchestra Now. *Endless Shout*. New York, NY: Tzadik, 2000.

Mitchell, Roscoe and David Wessel. *Contact*. Paris: Rogue Art, 2009.

Smith, Wadada Leo and Anthony Braxton. *Organic Resonance*. New York, N.Y: Pi Recordings, 2003.

Teitelbaum, Richard, Anthony Braxton and George Lewis. *Concerto Grosso for Human Concertino and Robotic Ripieno*. Therwil, Switzerland: Hat Hut Records, 1988.

APPENDIX 2: THE SOFTWARE PACKAGE

The IMP software package is archived at http://research.jeffalbert.com/imp. The most recent version will be available, along with any other pertinent information.

APPENDIX 3: MUSICIAN SURVEYS

The musicians who participated in the beta testing of IMP received the following consent script and verbally agreed for their names and comments to be used in this document.

Consent Script which includes description of project

"Computer Improvising Agent" is a project by Jeff Albert (jalber4@lsu.edu, 985.966.6093). The large scale goal of the project is to develop a computer program that will be an improvising agent in musical situations. The goal of this portion of the project ("Interviews with Improvisers"), is to learn about how improvisers listen during an improvised performance, and what audio features or musical events affect an improviser's musical decisions. The format of the the interviews with improvisers is as follows: a brief improvised musical duo performed by Jeff and the interviewee, which will be recorded, then Jeff and the interviewee will listen to the recording and discuss their musical decision making processes as they apply specifically to that musical situation, as well as in more general terms.

The participants are selected based on their established expertise in the field of improvised music. There is no compensation for participation, and no penalty for lack of participation. The recorded portions of the interviews (both musical and conversational) will likely be archived on the world wide web in mp3 format, and the interview portions will be transcribed and stored in text format as well. If you do not wish to have your interview posted online, please notify Jeff. By participating in this interview, you agree to be quoted in any publications that may result from this research. If you have any questions, please contact Jeff or, the LSU Institutional Review Board (irb@lsu.edu, or 225.578.8692).

Study Exempted By:
Dr. Robert C. Mathews, Chairman
Institutional Review Board
Louisiana State University
203 B-1 David Boyd Hall
225-578-8692 | www.lsu.edu/irb
Exemption Expires: 7-27-2014

BW

IMP Musician Survey

Respond to the following statements on a scale of 1 - 5 with 1 being strongly disagree and 5 being strongly agree.

1- What I played influenced what IMP played.

3

2 - What IMP played influenced what I played.

5

3 - There was musical communication between IMP and myself.

1

4 - Playing with IMP was a musical experience.

1

5 - I would like to hear others play with IMP.

1

6 - Please make any additional comments you would like.

It seemed clear that there was some level of response from the computer to me, but it was superficial at best. I tried as best as I could to respond musically to it, but it seemed like a fruitless endeavor.

Have to be honest, it wasn't a musical experience, at least according to my aesthetic.

MARK

IMP Musician Survey

Respond to the following statements on a scale of 1 - 5 with 1 being strongly disagree and 5 being strongly agree.

1- What I played influenced what IMP played.

2

2 - What IMP played influenced what I played.

3

3 - There was musical communication between IMP and myself.

3

4 - Playing with IMP was a musical experience.

2

5 - I would like to hear others play with IMP.

4

6 - Please make any additional comments you would like.

RAY

IMP Musician Survey

Respond to the following statements on a scale of 1 - 5 with 1 being strongly disagree and 5 being strongly agree.

1- What I played influenced what IMP played. 4

2 - What IMP played influenced what I played. 5

3 - There was musical communication between IMP and myself. 5

4 - Playing with IMP was a musical experience. 5

5 - I would like to hear others play with IMP. 5

6 - Please make any additional comments you would like.

I would have liked more 'real time' interaction with the program (I understand it may have been pre-loaded with input from an earlier session), though it did listen to me and came up with some interesting musical suggestions. I did have the overall feeling that we were 'communicating'.

RICK TROLSEN

IMP Musician Survey

Respond to the following statements on a scale of 1 - 5 with 1 being strongly disagree and 5 being strongly agree.

1- What I played influenced what IMP played.

2

2 - What IMP played influenced what I played.

3

3 - There was musical communication between IMP and myself.

1

4 - Playing with IMP was a musical experience.

5

5 - I would like to hear others play with IMP.

5

6 - Please make any additional comments you would like.

(I) IT WOULD BE INTERESTING TO SEE (HEAR) MORE DEVELOPMENT OF THIS PROGRAM.

JANNA

IMP Musician Survey

Respond to the following statements on a scale of 1 - 5 with 1 being strongly disagree and 5 being strongly agree.

1- What I played influenced what IMP played.

absolutely 5

2 - What IMP played influenced what I played.

absolutely 5

3 - There was musical communication between IMP and myself.

Sort of. In some ways yes. 3

4 - Playing with IMP was a musical experience.

Yes, 4

5 - I would like to hear others play with IMP.

Yes. 4

6 - Please make any additional comments you would like.

When I play with others, we communicate in more ways - eye contact, body movement, facial expression, along with musical cues. I noticed that when I played certain notes for a while, IMP played the same note(s). I think musicians would tend to play other notes that go with what I played. I was a bit annoyed at IMP for playing my note. ->

But I was pleasantly surprised by how much it did respond to me.

Also, when I play with others, sometimes they ~~other~~ decline to play what I'm doing, and go in another direction that is cool - and I like that. IMP seemed more cooperative with me. It also seemed to have somewhat of a "short attention span" - perhaps because it seemed to respond to me a lot.

IMP Musician Survey 19 Feb 2013

Respond to the following statements on a scale of 1 - 5 with 1 being strongly disagree and 5 being strongly agree.

1- What I played influenced what IMP played. 4

2 - What IMP played influenced what I played. 5

3 - There was musical communication between IMP and myself. 3

4 - Playing with IMP was a musical experience. 2 1/2 (or a strong 2)

5 - I would like to hear others play with IMP. ~~6~~

6 - Please make any additional comments you would like, especially as they mat pertain to differences from when you played with IMP on January 31.

I found the computer to be generally more pliable and responsive, particularly over the first 2 minutes. There still appears to be a significant lag between what I play and when it responds musically, but I was able to adjust for that: if I played an idea and then gave the computer 5-10 secs to respond, that seemed to work.

Still not a compelling musical experience — for that the computer would need quicker responses, some stronger ideas on

range, dynamics, and either harmony or at least the ability to discern pitch groups from the other improviser.

Then again, if you told me there actually was no difference between the two computer modules and the difference was me and my expectations... I'd probably believe you.

JANNA

IMP Musician Survey 19 Feb 2013

Respond to the following statements on a scale of 1 - 5 with 1 being strongly disagree and 5 being strongly agree.

1- What I played influenced what IMP played.

4

2 - What IMP played influenced what I played.

5

3 - There was musical communication between IMP and myself.

4

4 - Playing with IMP was a musical experience.

5

5 - I would like to hear others play with IMP.

5

6 - Please make any additional comments you would like, especially as they mat pertain to differences from when you played with IMP on January 31.

It is clear that IMP was not corresponding so exactly to my playing as it did last time. Yet, I thought that what it played fit with what I did. I thought it responded more "artistically" to my input.

MCGRAIN

IMP Musician Survey 19 Feb 2013

Respond to the following statements on a scale of 1 - 5 with 1 being strongly disagree and 5 being strongly agree.

1- What I played influenced what IMP played.

YES

2 - What IMP played influenced what I played.

YES

3 - There was musical communication between IMP and myself.

YES

4 - Playing with IMP was a musical experience.

YES

5 - I would like to hear others play with IMP.

SURE

6 - Please make any additional comments you would like, especially as they mat pertain to differences from when you played with IMP on January 31.

APPENDIX 4: IRB EXEMPTION

Application for Exemption from Institutional Oversight

Unless qualified as meeting the specific criteria for exemption from Institutional Review Board (IRB) oversight, ALL LSU research/projects using living humans as subjects, or samples, or data obtained from humans, directly or indirectly, with or without their consent, must be approved or exempted in advance by the LSU IRB. This Form helps the PI determine if a project may be exempted, and is used to request an exemption.

LSU
Institutional Review Board
Dr. Robert Mathews, Chair
131 David Boyd Hall
Baton Rouge, LA 70803
P: 225.578.8692
F: 225.578.6792
irb@lsu.edu
lsu.edu/irb

– Applicant,Please fill out the application in its entirety and include the completed application as well as parts A-E, listed below, when submitting to the IRB. Once the application is completed, please submit two copies of the completed application to the IRB Office or to a member of the Human Subjects Screening Committee. Members of this committee can be found at http://www.lsu.edu/screeningmembers.shtml

– A Complete Application Includes All of the Following:
- **(A)** Two copies of this completed form and two copies of part B thru E.
- **(B)** A brief project description (adequate to evaluate risks to subjects and to explain your responses to Parts 1&2)
- **(C)** Copies of all instruments to be used.
  *If this proposal is part of a grant proposal, include a copy of the proposal and all recruitment material.
- **(D)** The consent form that you will use in the study (see part 3 for more information.)
- **(E)** Certificate of Completion of Human Subjects Protection Training for all personnel involved in the project, including students who are involved with testing or handling data, unless already on file with the IRB. Training link: (http://phrp.nihtaining.com/users/login.php.)
- **(F)** IRB Security of Data Agreement: (http://www.lsu.edu/irb/IRB%20Security%20of%20Data.pdf)

1) Principal Investigator: Jeffrey V. Albert **Rank:** graduate student

Dept: Music **Ph:** 985-966-6093 **E-mail:** jalber4@lsu.edu

2) Co Investigator(s): please include department, rank, phone and e-mail for each

IRB# E5579 LSU Proposal #
⊗ Complete Application
⊗ Human Subjects Training on file

3) Project Title: Computer Improvising Agent

Study Exempted By:
Dr. Robert C. Mathews, Chairman
Institutional Review Board
Louisiana State University
203 B-1 David Boyd Hall
225-578-8692 | www.lsu.edu/irb
Exemption Expires: 7-27-2014

4) Proposal? (yes or no) no **If Yes, LSU Proposal Number**

Also, if YES, either
- ○ **This application completely matches the scope of work in the grant**

OR
- ○ **More IRB Applications will be filed later**

5) Subject pool (e.g. Psychology students) IMPROVISING MUSICIANS

*Circle any **"vulnerable populations" to be used:** (children <18; the mentally impaired, pregnant women, the ages, other). Projects with incarcerated persons cannot be exempted.

6) PI Signature [signature] **Date** 7 JULY 11 (no per signatures)

** **I certify my responses are accurate and complete.** If the project scope or design is later changes, I will resubmit for review. I will obtain written approval from the Authorized Representative of all non-LSU institutions in which the study is conducted. I also understand that it is my responsibility to maintain copies of all consent forms at LSU for three years after completion of the study. If I leave LSU before that time the consent forms should be preserved in the Departmental Office.

Screening Committee Action: Exempted ✓ Not Exempted ___ Category/Paragraph 2

Reviewer Mathews **Signature** [signature] **Date** 7/28/11

VITA

Jeff Albert is a musician, trombonist, improviser, music technologist, and composer. He is an Extraordinary Assistant Professor of Music Industry Studies at Loyola University New Orleans, and completed a PhD in Experimental Music and Digital Media at Louisiana State University, where his teachers include Stephen David Beck and Jesse Allison. He holds degrees from Loyola University - New Orleans, and the University of New Orleans, and has served on the faculty of Xavier University of Louisiana and the University of New Orleans. Jeff's areas of research include the intersections of improvisation and technology, and performance paradigms for live computer music.

Jeff was named a Rising Star Trombonist in the 2011 & 2012 Downbeat Critics Polls, and performs regularly in the New Orleans area, and throughout the US and Europe. In 2013, the Paris based record label Rogue Art released his CD, *The Tree on the Mound,* that features Kidd Jordan, Hamid Drake, and Joshua Abrams. In addition to leading the Jeff Albert Quartet, Jeff is a member of Hamid Drake's Bindu-Reggaeology band, and co-led the Lucky 7s with fellow trombonist Jeb Bishop. Jeff has performed with many great improvisers, including Georg Graewe, Tobias Delius, Dave Rempis, Jeff Parker, and Wolter Wierbos. He has been a member of the bands of New Orleans greats George Porter and Wardell Querzergue, backed artists like Stevie Wonder and Bonnie Raitt, and performed with the Louisiana Philharmonic Orchestra, and the New Orleans Opera.

Jeff is the founder and curator of the Open Ears Music Series, and writes the blog Scratch My Brain.