2012-05-07

Blue Skies: An Interactive Soundscape Composition

Daniel P. Dickinson

University of Miami, d.dickinson1@umiami.edu

Follow this and additional works at: https://scholarlyrepository.miami.edu/oa_dissertations

Recommended Citation
https://scholarlyrepository.miami.edu/oa_dissertations/762

This Open access is brought to you for free and open access by the Electronic Theses and Dissertations at Scholarly Repository. It has been accepted for inclusion in Open Access Dissertations by an authorized administrator of Scholarly Repository. For more information, please contact repository.library@miami.edu.
UNIVERSITY OF MIAMI

BLUE SKIES: AN INTERACTIVE SOUNDSCAPE COMPOSITION

By

Daniel P. Dickinson

A DOCTORAL ESSAY

Submitted to the Faculty of the University of Miami in partial fulfillment of the requirements for the degree of Doctor of Musical Arts

Coral Gables, Florida

May 2012
A doctoral essay submitted in partial fulfillment of the requirements for the degree of Doctor of Musical Arts

BLUE SKIES: AN INTERACTIVE SOUNDSCAPE COMPOSITION

Daniel P. Dickinson

Approved:

Dennis Kam, D.M.A.
Professor of Music Theory and Composition

Terri A. Scandura, Ph.D.
Dean of the Graduate School

Charles Norman Mason, D.M.A.
Associate Professor of Music Theory and Composition

Lansing McLoskey, Ph.D.
Associate Professor of Music Theory and Composition

Colby Leider, Ph.D.
Associate Professor of Music Engineering
Blue Skies is a site-specific interactive soundscape composition that integrates live musicians and algorithmic improvisation into a sound installation that reacts to changing environmental conditions. The composition provides a template for multiple instances of the piece to be composed in the future for different sites. It includes an improvising software and hardware system called the Motivator that generates music based on parameters established as part of the composition, reacting in specified ways to changes in wind speed, light level, and other factors at the installation site. It also suggests a model for producing instructional, graphic scores that can be interpreted by live musicians as well as being translated into a format the computer system can understand. Included is a version of the composition composed specifically for presentation at the University of Miami campus in Coral Gables, using live musicians and field recordings made in and around the campus.

An accompanying paper contains a discussion of the components of the piece and the composer’s intentions from several perspectives, including issues of interactivity between a musical composition and its performers, audience, and location; and issues of composition for spaces outside of traditional performance venues. The paper also presents brief considerations of work by several influential composers whose work is related to Blue Skies, including John Cage, Max Neuhaus, and John Luther Adams.
ACKNOWLEDGEMENTS

Many people have made valuable contributions to my work at the University of Miami that has led to this composition. First, I would like to acknowledge my wonderful committee. To Dennis Kam, thank you for encouraging me to further pursue improvisational forms and for all your guidance in helping me grow as a composer. I have enjoyed our many conversations over the years and have grown immensely with your guidance. To Colby Leider, thank you for introducing me to the field of acoustic ecology and for helping me find a way to integrate my engineering experience into my music. To Charles Mason, thank you for your helpful insights over the course of this project, especially as I explored sound installations for the first time. To Lansing McLoskey, thank you for your always candid and constructive advice over the years. To Juan Chattah, who served temporarily on my proposal committee, thank you also for your helpful insights on the theoretical aspects of this piece.

I would also like to thank my colleagues Matthew Taylor, Liza Seigido, and Bruce Johnston, with whom I’ve enjoyed many hours of live improvisation. That ongoing dialogue has contributed greatly to the current work and my own growth as a musician. Also thanks are due to Sophia Serghi who really encouraged me to pursue graduate studies in the first place, and to the rest of the composition faculty at the University of Miami: Paul Wilson, Bob Gower, and Scott Stinson, all of whom have taught me a lot about composition and teaching.

Finally, thank you to my sons, Maxwell and Isaac; my parents, George and Phyllis Dickinson; and especially my wonderful wife Roxane Pickens, who has been my best friend and supporter for 20 years and counting.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>LIST OF FIGURES</td>
<td>v</td>
</tr>
<tr>
<td>LIST OF MUSICAL EXAMPLES</td>
<td>vi</td>
</tr>
<tr>
<td>LIST OF TABLES</td>
<td>vii</td>
</tr>
<tr>
<td>1 INTERACTIVE SOUNDSCAPE COMPOSITION</td>
<td>1</td>
</tr>
<tr>
<td>2 BLUE SKIES</td>
<td>6</td>
</tr>
<tr>
<td>Component Overview</td>
<td>7</td>
</tr>
<tr>
<td>Algorithmic Model</td>
<td>9</td>
</tr>
<tr>
<td>Computer System</td>
<td>14</td>
</tr>
<tr>
<td>University of Miami March 2012 Instance</td>
<td>22</td>
</tr>
<tr>
<td>3 COMPOSITION AND THE PERFORMANCE SPACE</td>
<td>30</td>
</tr>
<tr>
<td>Audience Behavior and the Aural Environment of the Performance</td>
<td>30</td>
</tr>
<tr>
<td>John Cage and Max Neuhaus</td>
<td>31</td>
</tr>
<tr>
<td>4 EXPANSION OF INTERACTION THROUGH COMPOSITION</td>
<td>35</td>
</tr>
<tr>
<td>Expansion of Interaction with the Environment</td>
<td>35</td>
</tr>
<tr>
<td>Level of Human (Audience) Interaction</td>
<td>40</td>
</tr>
<tr>
<td>Complexity of Interactions</td>
<td>42</td>
</tr>
<tr>
<td>5 REFLECTIONS AND FUTURE WORK</td>
<td>45</td>
</tr>
<tr>
<td>BIBLIOGRAPHY</td>
<td>49</td>
</tr>
<tr>
<td>APPENDIX A: SCORE FOR UM2012 EDITION</td>
<td>52</td>
</tr>
<tr>
<td>APPENDIX B: SCORE FOR DEMO EDITION</td>
<td>57</td>
</tr>
<tr>
<td>APPENDIX C: MAX PATCHES</td>
<td>62</td>
</tr>
</tbody>
</table>
LIST OF FIGURES

Figure 2.1. Blue Skies provides a template for generating various compositions, each of
which is intended for a particular site.................................................................7

Figure 2.2. Components of the Blue Skies template...........................................8

Figure 2.3. Overview of the computer system for Blue Skies. ..............................15

Figure 2.4. Data flow among Motivator components...........................................17

Figure 2.5. Motivator components responding to a single light-level change. ..........18

Figure 2.6. A slightly more complex Motivator with two sensors and two instruments...19

Figure 4.1. Behavioral guidance before and after the introduction of a building. On the
left, the person is free to move about the central area of the empty lot. On the
right, the person’s behavior is guided by the construction of the building. ....38
LIST OF MUSICAL EXAMPLES

Example 1.1. The “performers” (sound producers) shape the soundscape with their sonic material, prompted by environmental conditions. ..................................................2

Example 1.2. In Blue Skies, we’ve added musicians and algorithmic improvisers to our model of soundscape generation. ..................................................3

Example 2.1. State machine from a simple Blue Skies composition designed for a demonstration lecture. .......................................................................10

Example 2.2. Detail of the “Start,” “Day,” and “Night” states from Example 2.1. ........11

Example 2.3. Detail of the “Day 2”, “Night 2”, and “Wait for Day” states. ...............12

Example 2.4. Melody of the University of Miami clock tower. ..................................24

Example 4.1. Excerpt from Reactions, demonstrating composed responses to environmental cues. .................................................................36
LIST OF TABLES

Table 3.1. Differences between the traditional concert hall and outdoor experiences.....31

Table 4.1. The role of each contributory factor in the piece. ........................................40
Chapter 1

INTERACTIVE SOUNDSCAPE COMPOSITION

In his book *The Soundscape*, R. Murray Schafer announces his intention “to treat the world as a macrocosmic musical composition.”¹ The soundscape—the collection of sounds that is heard in a particular place—presents an interesting example of an ever-evolving piece of music with no clear beginning or end, and no composer. Schafer’s statement suggests that it is the observer who designates a set of sounds to be a “composition,” thus obviating the need for a composer. In the case of the soundscape, an intriguing world of sound already exists everywhere we go, some of it produced by humans, but much of it not.

However, we can imagine an ensemble of improvising performers who compose the soundscape. In a traditional ensemble, the performer is the person who serves as a catalyst to begin the vibration of some material (a reed, a string, a membrane, or a speaker cone) that produces a sound. In a soundscape, these performers need not be trained musicians or even humans. They are simply whatever is producing the sound. For example, when the wind blows, it moves leaves in the trees that create a sound, it pushes waves to break against the shore, and so forth. Birds sing, crickets chirp, fans hum, people talk and run lawn mowers. All of these contribute to the orchestration of the soundscape.

More precisely, we can say that the “performers” are *improvising* the soundscape. The soundscape is not pre-composed. However, just as in most improvisation, a certain vocabulary is in use, and some elements might be considered “pre-composed.”

¹ R. Murray Schafer, *The Soundscape: Our Sonic Environment and the Tuning of the World* (Rochester,
example, a mourning dove has a standard call that varies slightly from dove to dove and place to place, but certainly can be seen as variations on a particular motivic idea. In this sense, we can think of this idea as “composed” material, which the dove then uses as raw material for an improvisation. In a similar sense, we might think of the sound of a lawn mower, the chatter of a crowd of college students, or the hum of an air conditioner as emerging spontaneously from a core of sonic material: the timbre of a motor, or the fluctuating frequencies of human speech. This model is demonstrated in Example 1.1.

Example 1.1. The “performers” (sound producers) shape the soundscape with their sonic material, prompted by environmental conditions.

Into this ongoing performance, we can introduce our own sounds, altering the soundscape. This paper describes the piece Blue Skies, which realizes a particular approach to this alteration. Blue Skies includes a computer-based improviser that produces musical material using various algorithms, employing sonic material culled from a combination of traditional musical instruments and field recordings from the installation site. Changes in the improvisation are driven in part by sensors that detect changes in the environment, such as wind speed and light levels. The piece also includes an instructional score that provides performing musicians with their own algorithms for
improvising, including responding to environmental conditions. This gives us the overall compositional model shown in Example 1.2.

Example 1.2. *Blue Skies* adds musicians and algorithmic improvisers to the previous model of soundscape generation.

An important desired effect of this piece is to increase the participants’ awareness and sense of connectedness to everything else around them. By noticing the sonic changes that are produced by simple changes such as the wind, listeners may begin to perceive these changes in a new light. By hearing a sound that they hear routinely in their own environment used somewhat differently in a piece of music, listeners may begin to hear the original sound as something musical, so that the next time they hear it they are
more conscious of it, the way they might immediately recognize a familiar melody from a pop song.

A similar desire to heighten connection is part of John Luther Adams’ stated motivation behind *The Place Where You Go to Listen*, his installation at the Museum of the North in Fairbanks, Alaska. In *The Place*, Adams uses synthesized sounds, but changes in the sound are affected by a number of environmental factors, including light level, but also including seismic activity and solar radiation (which causes the Aurora Borealis). Adams writes:

As a composer, I believe that music can contribute to the awakening of our ecological understanding. By deepening our awareness of our connections to the earth, music can provide a sounding model for the renewal of human consciousness and culture.\(^2\)

In essence, the goal is a sort of “transmodal” approach, in which observers are given a new perspective on familiar material by transcribing it into a different context or medium. Anyone living on the West Coast of the United States has probably experienced the tremors and earthquakes that accompany seismic activity, but by sonifying these effects in his piece, Adams presents them in an unfamiliar format, triggering a new awareness in the listener. At the same time, by magnifying tremors that might be too small to feel, Adams reminds the audience that there is a lot more to this activity than they might assume.

This paper presents *Blue Skies* and its relationship to these goals and aesthetic ideas. The next chapter will examine in detail the piece itself and the various components that comprise it. Subsequent chapters will discuss various aesthetic issues that arise from

this sort of composition, focusing on considerations of the performance space and the expansive possibilities for interactivity in the composition.
Chapter 2

BLUE SKIES

It is best to think of Blue Skies not as one composition designed to produce a particular sonic outcome, but rather as a collection of possible compositions, each of which contains certain specific components and follows a certain template. The components are: a site where the piece is to be performed; the Motivator computer program that improvises algorithmically; a score readable by this program that guides the algorithmic improvisation; various sensors that feed data about the environment into the computer program; sampled sounds used as sonic material by the Motivator; an ensemble of performers; and a score readable by the performers that guides their improvisation. These components will be present in every “instance” of the Blue Skies template. However, many of the details will vary and should be determined by the site. Figure 2.1 shows the relationship between this template, the various compositions that can be produced, and the sites. This chapter discusses these components in their general form as part of the template. It then describes the first instance of the piece, intended for performance at the University of Miami in Coral Gables, Florida, in April or May 2012. This particular instance will be referred to as UM2012.
Figure 2.1. *Blue Skies* provides a template for generating various compositions, each of which is intended for a particular site.

**Component Overview**

The components of the *Blue Skies* template are shown in Figure 2.2. As shown in the diagram, we can conceptualize the various components as two subgroups: the computer system and the live performer system. Both of these systems are then related to the site itself. The site is the first component that we will examine.
The site is an important consideration. First, the site determines the environmental cues that trigger responses from both the computer and the performers. Second, at least some of the sonic material used by the Motivator should be derived from sounds heard at the site. Third, the presence of existing sounds in the soundscape at the site may affect compositional decisions that go into producing the score. For the first version of Blue Skies, the site is located at the University of Miami campus in Coral Gables, Florida, USA. This site was chosen primarily because of its ease of access, but it also provides an interesting soundscape that blends plant, animal, and water sounds with mechanical and human sounds. More about the site is discussed below.

The second component is the Motivator computer program and its related components, sensors, and sonic material. At the time of this writing, this program is written in the Max visual programming language produced by Cycling ‘74 and run on an
Apple computer running Mac OS X. The sonic material consists of a combination of pre-packaged library sounds and sampled sounds that were recorded at the site. In preparing an instance of Blue Skies, the composer should always allow time to obtain field recordings of the site and turn these recordings into digital instruments that can be played by the Motivator. Currently, the system uses the commercially available sampling engine Kontakt 3 from Native Instruments. Specific sounds used in the UM2012 version are discussed below.

The sensors gather data about the environment and feed it into the Motivator software. Potentially, a wide range of sensors can be utilized, but the current system uses an anemometer to measure wind speed, a photosensor to measure ambient light levels, and a microphone to respond to certain sonic events. The anemometer and light sensor transmit current conditions wirelessly to the computer, which then responds to changes in conditions in a way that is determined by the electronic score. The sensors, the score, and other details of the computer system are discussed in subsequent sections.

The final components in the template are the live performers and the score that they follow. The instrumentation is unspecified by the template, and should be determined based on the particular instance of the composition.

**Algorithmic Model**

The algorithmic model for Blue Skies is based on a finite state machine. A finite state machine (or simply “state machine”) is a concept from computer science that describes the states in which a system can exist, the possible transitions between these states, and the inputs into the system that cause these state transitions. In Blue Skies, each state
represents a particular set of instructions for the performer. The inputs are weather events or other events that are likely to occur at the site.

Example 2.1 shows a state machine from a simple version of *Blue Skies* that was composed for a lecture demonstration. This portion of the score is to be played by a digitally sampled harp. The rectangles represent the states of the composition. The arrows represent transitions. The input that triggers a particular transition is represented either by text (such as “Too slow!”) or an icon, such as the “sun” icon, representing bright light, or the “sun-obscured-by-cloud” icon, indicating darkness or shadow. This particular example was composed to be performed in a classroom, and thus allows for only a simple opposition of “light” and “dark,” which was easily simulated in this case by turning the lights on and off at the switch.

Example 2.1. State machine from a simple *Blue Skies* composition designed for a demonstration lecture.

In this example, the performer begins in the “Start” state, which provides a scale and the instructions to compose three motives using the scale. The arrows pointing out of the “Start” box indicate two possible transitions to the next state. If the light level is high
enough, the performer should move to the “Day” state. If it is dark, the performer should move to the “Night” state. In the “Day” state, the performer is asked to repeatedly play the motives that were composed in the previous state. There is only one transition out of the “Day” state, triggered by the onset of relative darkness (such as by a cloud moving across the sun), at which point the performer moves into the “Night” state and is asked to improvise on two notes. (These first three states are reproduced in Example 2.2.)

Example 2.2. Detail of the “Start,” “Day,” and “Night” states from Example 2.1.
Example 2.3. Detail of the “Day 2”, “Night 2”, and “Wait for Day” states.

The performer stays in the “Night” state until the performance space becomes brightly lit again, at which point he or she moves into the “Day 2” state, shown close-up in Example 2.3. This state is very similar to the “Day” state in that the performer is asked to repeatedly choose and play motives that were composed in the “Start” state. However, here the performer is also asked to pay attention to the speed of the wind and adjust the tempo of the performance accordingly. A transition out of this state occurs when the area becomes dark once again, and the performer moves into the “Night 2” state. Here, the performer is again asked to improvise on two notes (the same two from the “Night” state). But this state also calls for the performer to gradually slow down.

The “Night 2” state has two possible transitions: the one on the left triggered by the return of bright light, or the one on the right triggered when the performer feels that she is playing as slowly as she wants to play (“Too slow!”). If the latter occurs before the former, then the performer moves into the “Wait for Day” state, simply doing nothing.
until the light level rises once again, at which point she returns to the “Day” state and the
process starts over. If the light level rises before the performer has gotten “too slow,”
then she simply moves directly back to the “Day” state. Once back in the “Day” state, the
cycle simply continues indefinitely. An alternate approach might eventually specify a
transition to an “End” state, but that was not included in this example.

We can make a number of observations from this example. First, within each
state, the amount of indeterminacy and improvisation is highly variable. We can prescribe
an exact series of notes for the performer, or we can simply provide instructions for
improvising. Second, we can specify triggers that cause transitions between states, but
there are two variables during performance: the actual point in time at which the trigger
occurs, if it occurs at all; and the subjective interpretation of the trigger. For example,
darkness as interpreted by a human performer is somewhat subjective. However, this
level of subjectivity is really not much different from standard variations in the
interpretation of tempo and dynamics in a more traditional piece. The variability of the
occurrence of the trigger is somewhat more interesting, as this can potentially lead to
significantly different performances. If we are performing the harp part shown in
Example 2.1 during the day, we might interpret “darkness” to be whenever the sun goes
behind a cloud. On a partly cloudy day with a fair amount of wind, the sun might be
popping in and out of cloud cover every few seconds, causing rapid transitions between
states. However, on a clear, sunny day, the sun might never go behind a cloud, and the
piece would essentially stay in a “Day” state until sundown. These consequences may or
may not be the desired effect, and they are aspects that need to be considering when
composing using this model.
A useful aspect of this approach to scoring is that the same score can be used for both human performers and the computerized improvisation system. The diagram can be read directly by the performer, but it needs to be converted into a different format for the machine. Nevertheless, the point of origin for both can be the same. The primary difference is that we end up providing more precise instructions to the computer, so that interpretive questions such as “light level” can be given in precise values. However, in the score, I have chosen to use more general terms, such as “brighter” or “darker.” In this way, the score is more understandable to people, and the translation into the computer-readable format becomes simply another aspect of interpretation.

The score draws on the fine tradition of instructional, event, and open form scores from composers such as George Brecht, John Cage, La Monte Young, and Terry Riley, or visual artists such as Sol LeWitt. Like 4’33”, or Brecht’s Flute Solo—consisting of two words: “disassembling” and “assembling”—the score does not clearly specify the passage of time, but it does divide the time span of the performance into well-defined situations.³ In this sense, it serves a similar function to a traditional score from, for example, late 18th-century Vienna. The score makes use of some traditional notation as well as invented graphical symbols to indicate certain gestures. Simple text instructions are also included to ensure a clear understanding of the graphics.

**Computer System**

Considerable effort went into developing a computer system designed to interpret these compositions. Figure 2.3 gives an overview of this system. In developing the system, I

---
³ A number of these works are discussed in Liz Kotz, “Post-Cagean Aesthetics and the Event Score,” *John Cage*, Julia Robinson, ed. (Cambridge, Massachusetts: MIT Press, 2011), 102–140.
used the Max visual programming language produced by Cycling ’74 (Max 6, precisely). For sound reproduction I used the Kontakt sampling engine from Native Instruments. Both of these are running on an Apple computer under Mac OS X. I chose Kontakt primarily because it is a powerful sampling engine and I already owned a copy. Really any sample playback mechanism would probably work fine. Separating the sampler from the algorithmic aspects of the Motivator helped during the implementation phase, because it further emphasized the degree to which the sampled sounds were just instruments that were being played by the Motivator, analogous to the instruments being played by the performers: raw materials capable of emitting a certain set of sounds.

Figure 2.3. Overview of the computer system for Blue Skies.

The motivation behind using Max is somewhat more complicated. Its main advantage for this project is that it provides a very simple set of tools for communicating with MIDI devices and for reading input over serial ports and over the network. This is a fairly significant bit of software infrastructure that is very easy to use in Max. However, a disadvantage of Max is that it can be quite complex to develop simple algorithms that are better suited to text-based programming language such as Java or C-Sharp.
The sensor mechanisms were developed using the Arduino open-source hardware platform. Arduino hardware can be purchased in several different formats at reasonable cost. Each Arduino model consists of a microcontroller with a large number of digital and analog inputs, to which sensors can be attached. A microcontroller is a microprocessor designed to be programmed for a single application and embedded in an appliance or other electronic device. Typically, dealing with microcontrollers is an experience unforgiving to novices, so the Arduino platform was developed to smooth over some of obstacles that a hobbyist would encounter. It includes software that makes it relatively easy to write programs and install them on the Arduino.

For this project, I am using an anemometer to measure wind speed, and an ambient light sensor to measure light levels. As shown in Figure 2.3, these sensors are connected to the Arduino, which processes the raw electrical signal and converts it into messages. The messages are then transmitted wirelessly back to the computer, where they are handled by the Max patch. Wireless transmission is handled by XBee devices, which utilize a protocol known as ZigBee intended for low bandwidth requirements and low power consumption.  

The Max patch consists of three main components: the Composer/Improviser, which is capable of composing motives and improvising according to specified parameters; the State Machine, which reads the computer-readable version of the score and uses it to instruct the Motivator; and the Sensor Monitor, which reads data from the sensors. Figure 2.4 shows the data flow among these components. This diagram is

---

4 The ZigBee protocol itself is built on the IEEE standard 802.15.4. For more information, see José Gutiérrez, et al, Low-Rate Wireless Personal Area Networks: Enabling Wireless Sensors With IEEE 802.15.4 (New York: IEEE Press, 2010). An excellent practical guide to setting up XBee hardware with the Arduino can be found in Tom Igoe, Making Things Talk (Beijing: O’Reilly Media, 2007).
simplified to only show one of each component, but in the implementation there are multiple instances: one Sensor Monitor for each sensor type, and one State Machine and one Composer/Improviser for each MIDI instrument.

Figure 2.4. Data flow among Motivator components.

The data flow in Figure 2.4 works as follows. First, sensor values are received by the Sensor Monitor. The value will indicate something like the actual wind speed or the ambient light level. The monitor then takes this data and uses it to produce sensor events. There are several possible events, but the most useful is the “range” event. This indicates that the data value falls within a particular, predefined range. For example, we might have three light-level ranges: 0% to 30%, 30% to 70%, and 70% to 100%. These would be ranges 1, 2, and 3, respectively. The State Machine then listens for sensor events. In the Score, a transition may be triggered by a particular sensor event. Continuing with our light-level example, the Score might indicate that when we are in State A, and we receive a “light range 2” event, we should transition to State B. The State Machine processes the event and changes its own state as appropriate. Then, it reads the commands that are listed in the new state, passing them to the Composer/Improviser.

The Composer/Improviser executes the commands. If a “play” command is given, the Composer/Improviser will begin generating MIDI notes and sending them to its MIDI device. It will also generate musical events, which are passed back to the State Machine.
and to other Composer/Improviser instances. These musical events include aspects of the music that might trigger a state change, such as thresholds of tempo or pitch that have been crossed. In this way, the rest of the system is able to respond to the notes that are generated by each Composer/Improviser, and the system can handle triggers such as “Too slow!” from the score excerpt shown earlier in Example 2.1.

In addition to sensor changes triggering state changes, the Composer/Improviser can also be configured to listen directly to sensor events. These events can be tied to a particular parameter, such as pitch or timing, allowing environmental changes to create more subtle variations in the music.

Figure 2.5. Motivator components responding to a single light-level change.

Figure 2.5 is a similar diagram to Figure 2.4, but showing specific messages. Here, the light level is passed in from the sensors. The new light level is 412, which is on a scale from 0 to 1023. The Sensor Monitor looks the level up in its range table and generates a “light range 2” message. The State Machine looks up that message based on its current state, and sees that a transition is indicated to state “Day.” It switches to state “Day” and transmits the commands in that state, which are “choose motive” and “play.” The Composer/Improviser chooses one of its previously composed motives and begins to play it, sending note-on and note-off messages over its MIDI channel.

As mentioned earlier, in most cases there will be multiple Sensor Monitors, multiple State Machines, and multiple Composer/Improvisers. Figure 2.6 provides an
example. Here, the two Sensor Monitors would each be attached to one particular sensor, such as a light meter and an anemometer. Each Composer/Improviser is sending MIDI data to its own instrument in the sampler. Notice in this example how each State Machine is capable of responding to all sensor events. Also, notice how musical events from the Composer/Improviser 2 are passed back not only to its own state machine, but to State Machine 1 as well. This cross-communication will allow Instrument 1 to “listen” and respond to Instrument 2. (Instrument 2 can also “listen” to Instrument 1; this arrow has been omitted to simplify the diagram.)

Figure 2.6. A slightly more complex Motivator with two sensors and two instruments.

The development of a computerized improviser has many important precedents in computer music. Early efforts automated the composition process apart from performance. Lejaren Hiller and Leonard Isaacson in 1957 produced the *Iliac Suite for String Quartet*—what Hiller called “the first substantial piece of music produced with a computer”—at the University of Illinois at Urbana. Based on information theory, the piece used a “generate-and-test” model, in which a number representing a musical parameter (pitch, duration, etc.) is randomly selected, and then tested against criteria established by the composers. A series of these parameters was generated by the
program, printed out, and manually converted into musical notation to be played by
performers. This approach was similar to Cage’s chance processes, but relocated into
into the computer, and further filtered by the composer’s criteria.  

From 1967 to 1969, Hiller worked with Cage on the piece *HPSCHD*, which
started as a commission for harpsichord but evolved into a complex multimedia piece for
tapes, keyboards, and videos. The *HPSCHD* software improvised from various scales and
wrote these improvisations to tape. One of the tapes was transcribed for keyboard and
included as one of the seven solos from which the harpsichordists may choose to
perform. As Hiller wrote:

> Each tape, once it is started, must run through to its end. Meanwhile, other
tapes might be started. Each keyboard performer should select and play
through one of the seven versions for keyboard to the end…Then he can
get up, go out, smoke a cigarette, have a drink—whatever he wants—and,
if he feels like it, come back and play the same version again or play
another.  

*HPSCHD* contains many of the elements used by *Blue Skies*: constrained
randomness used to generate pitch material; open-ended performance times; several
options from which the performer can choose; computer-generated music combined with
live performers. It was not the first to incorporate any of these elements; however, it
stands as an important illustration of many of the possibilities engendered by these
methods.

The technical nature of the early computer systems employed in the 1960s at
Illinois helped prevent Hiller and Isaacson from generating improvisations in real-time,

---

5 Joel Chadabe, *Electronic Sound: The Past and Promise of Electronic Music* (Upper Saddle River, New
but by the time George Lewis began his explorations in the 1980s, he was able to leverage the capabilities of the personal computer to produce a real-time improvising software package called *Voyager*. This program responded to a live performer (playing through a microphone attached to an event and pitch detector) to produce a complex performance that reacted to the musician while maintaining its own autonomous direction. Lewis explains:

My analysis of *Voyager* as an interactive computer music system uses Robert Rowe's taxonomy of "player" and "instrument" paradigms, although these two models of role construction in interactive systems should be viewed as on a continuum along which a particular system's model of computer-human interaction can be located. In Rowe's terms, *Voyager* functions as an extreme example of a "player" program, where the computer system does not function as an instrument to be controlled by a performer.

I conceive 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.

Although it does not engage in direct interaction with the performer, in the same manner as *Voyager*, the Motivator follows a similar “player” paradigm, with the computer acting as a wholly autonomous agent, reacting to outside stimuli and following the score in the same manner as the live performers.

Many other composers and engineers have developed computer systems designed to improvise based on interactions with external stimuli. As in Lewis’s *Voyager*, the stimuli are typically provided by a performing musician instead of naturally occurring events, but the basic underlying principle is the same: the computer responds to stimuli by generating its own music in a manner that is often unpredictable. Also like *Voyager*,

---

much of this work was motivated by the increased and more accessible computing power that became available in the early 1980s. Larry Polansky, Phil Burk, and David Rosenboom developed an object-oriented programming language called HMSL (Hierarchical Music Specification Language) to facilitate the creation of interactive music software. William Walker developed a programming framework for building computer improvisers called ImprovisationBuilder, focused within the jazz idiom. Joel Chadabe’s $M$ software offered a graphical user interface for setting ups similar responsive improvisations. Developed for the fledgling Apple Macintosh starting in 1985, $M$ is especially notable for providing a relatively easy-to-use tool for establishing the parameters within which the computer will produce music. This process is analogous to the pre-performance preparation in Blue Skies in which the written score is translated into a computer-readable state machine. However, $M$’s user interface allows the composer more flexibility to “play” with the parameters, offering a somewhat more intuitive process.\textsuperscript{8}

**University of Miami March 2012 Instance**

As discussed earlier in this paper, Blue Skies provides a template for generating many actual compositions, or “instances” of the template. For this project, I have composed two such instances. The first is the demonstration discussed above and excerpted in Example 2.1. This version was developed for a lecture demonstration in an interior classroom. The second instance was designed for presentation outdoors at the University of Miami

campus in Coral Gables, Florida. In this section, I will discuss this iteration in more detail.

The site, in the music school, presents an interesting soundscape, recently transformed by the ongoing construction of a new student center adjacent to the school of music. Typical sounds heard around the school are the coo of the mourning dove, the dry rustle of palm leaves blowing in the wind, and the sound of a large fountain in the middle of Lake Osceola, located at the center of campus adjacent to the music school. Occasionally, the very loud call of a small flock of macaws can be heard flying by. There are of course the various sounds of people in a semi-urban environment, including conversation, footsteps, and the sounds of car engines. Many drone sounds can be heard as well. In addition to the aforementioned fountain, the most prevalent of these sounds is the drone of air conditioners. In fact, a very interesting sound walk can be taken through campus at night, listen to the changing pitches and timbres of the various central air conditioning units as you walk among the buildings. Sometimes, the air conditioners are accompanied by an additional chorus of crickets. In addition to air conditioners, the sound of the two-cycle engine used in weed cutters, leaf blowers, and other small, powered landscaping equipment is extremely common during the day. While the sound of air conditioners typically has a pitch accompanied by a lot of white noise, the landscaping engines often have a very well-defined pitch, albeit one that changes and glissandos as the engine starts and stops or strains a bit under a workload.

As of this writing, the construction project is now a major contributor to the soundscape. As anyone who has spent much time near a construction site (at least in the United States) will know, a particularly pungent sound product is the warning beep of
construction equipment in reverse. This can be tracked at different speeds and pitches. And although the regularity and timbre of the sound make it arguably the most annoying of construction noises (indeed, it is annoying by design so that it is clearly audible to the workers), it introduces a certain tonality into the soundscape by virtue of its clear pitch. It shares this characteristic with the droning weed-cutting equipment.

In addition to the warning beeps, the construction site also produces a lot of large engine noises (with a lower and less-clearly-defined pitch than their two-stroke handheld cousins), as well as the sounds of digging and hammering and so forth. A particularly interesting sound heard occasionally at this site is the rhythmic crunch of a very large drill being used to install support columns for the building.

Another prominent sound on campus is that of a clock chime located in the main library. This chime tolls every half-hour during the day, and up until about 9:00 in the evening. The melody is a very common four-note bell melody in G major consisting of G, A, B, and D, shown in Example 2.4. After the melody the hour is chimed with the chord shown in the example between the repeat signs.

Example 2.4. Melody of the University of Miami clock tower.

In addition to their acoustic properties, all of these sounds have interesting sociological connotations. In particular, many of them highlight the relationship between humans and the rest of nature in the subtropical South Florida environment. The air conditioners and lawn tools emphasize the desire to control this environment by changing the prevailing temperature (hot) and the growth tendencies of flora (abundant). The
occasional macaws are a non-native group that most likely escaped from a local animal park during a storm. Their loud arrival signifies an ironic triumph: they would not be here if they (or their ancestors) had not been imported, but now they have become a welcome part of the local wildlife. The macaws have thus come full circle from “wild” to domesticated, and back again to wild.

For this *UM2012* iteration of *Blue Skies*, I made some field recordings around campus. These recordings were edited and imported into the Kontakt sampling engine. In that software, the samples were further edited and manipulated to create digital instruments that could be played via MIDI by the Motivator software (that is, the Max patch). Several principles guided the creation these instruments. First, I wanted to retain enough of the characteristics of each sound that its source would be easily recognizable. This approach is in contrast to another interesting and common electronic music strategy in which a sample is manipulated in many ways, often with the purpose of creating something completely new from a familiar source. I wanted the sounds to be recognizable so that their connection to the environmental soundscape would be clear. In this way, the piece and the existing soundscape engage in an intertextual dialog centered around their shared sonic vocabulary, a dialog which may continue in the minds of listeners long after the piece has stopped playing.

The second decision regarding the sonic material was to maintain a sense of pitch. I decided to treat most of the sounds as pitched sounds, and transpose them across the range of the MIDI keyboard. This mapping was done in different ways for different instruments. For example, the “dove” instrument takes advantage of one of Kontakt’s granular time stretching algorithms (“time machine 2”) to shift the pitch without
changing the speed of the sample. However, the “clock chime” instrument employs the standard algorithm of simply slowing down or speeding up playback of the sample to change the pitch, using several different samples on different ranges of the keyboard.

Another interesting characteristic of the treatment of these sounds relates to background noise. A small field recorder was used to record the sounds, and most of the recordings contained significant background noise. For the dove sound, which has fairly focused pitch and narrow range, a simple multiband compressor proved effective in removing this broadband noise. However, this method did not work as well for the clock chime, so a noise reduction algorithm found in the Audacity sound editor software was used instead. This algorithm eliminated the noise, but also left a considerable amount of artifacts that resemble the bubbling sounds of an old analog synthesizer. These artifacts were interesting in themselves, so I decided to simply leave them in. In the end, the clock chime ended up being one of the most manipulated sounds, making use of looping and reverse playback to shape the sound over time.

In the early stages of planning *Blue Skies*, I had intended to only use sampled sounds acquired from the local soundscape. But while developing the Motivator software in Max, for testing purposes I used sampled harp and clarinet instruments that are included with the Kontakt software. The interaction of these sounds with the Motivator software proved so compelling that I decided to utilize these sounds in the final piece as well. Part of the appeal is the frequent use of pitch bends. The Motivator specifies pitches in frequencies, which are then converted into MIDI pitches along with a MIDI pitch-bend
value if the frequency does not fall neatly onto a twelve-tone equal-tempered note. The pitch-bend parameter provides one approach to allowing off-the-shelf, equal-tempered instruments such as a harp to play microtonal scales. Because I was making use of microtonal scales in the composition, most notes required some pitch bend. Ideally, the pitch bend would be applied at the beginning of the note and would be maintained throughout, resulting in a clean, non-equal-tempered pitch. But because the pitch bend control changes the pitch for the entire instrument, rather than individually for each note, the sounding of two overlapping notes on the same instrument often results in a clearly audible glissando as the pitch bend message is sent. This is actually a very pleasant effect, and in particular gives the harp an unusual sound, uncharacteristic of the standard acoustic instrument.

Table 2.1 lists the digital instruments used in the piece. As mentioned earlier, most of these instruments are treated as pitched instruments and tuned roughly to the equal-tempered scale, but using the pitch-bend parameter to allow for additional microtonal notes. This emphasis on pitch has several interesting consequences. First, it allows the site-sampled instruments to coexist musically with the traditional instrument sounds. For example, at one point in the piece, the dove engages in a canon with the harp. Second, the pitch of the sampled instruments references the pitched sounds in the environment. Although composers and other musicians may be attuned to the pitches present in environmental sounds, using these sounds in an obviously musical context might help clue in other listeners to the musical sounds all around us.

---

9 This strategy assumes that the pitch-bend control is configured to change the pitch by a maximum of one semitone up or down.
Table 2.1. Digital Instruments Used in *Blue Skies: UM2012*.

<table>
<thead>
<tr>
<th>Instrument</th>
<th>Sound Sources</th>
<th>Instrument Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dove</td>
<td>Recorded call of a mourning dove</td>
<td>Sample was filtered to remove background noise, then transposed across the keyboard using a granular time-shifting algorithm supplied by Kontakt. Convolution reverb applied to the final sound to soften some harshness in the processed recording.</td>
</tr>
<tr>
<td>Clock Chime</td>
<td>Clock tower chimes</td>
<td>Sounds are modified, looped, and reversed to create a more complex version of the original. “Underwater” quality comes from the original noise removal processing, which left unusual artifacts.</td>
</tr>
<tr>
<td>Underwater</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Clock Chime</td>
<td>Clock tower chimes</td>
<td>Similar to Clock Chime Underwater, but with the ADSR envelope’s release time set much shorter to create a more percussive effect.</td>
</tr>
<tr>
<td>Short</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Construction</td>
<td>Beeping backup warning indicator, pile-driving drill, other motor sounds.</td>
<td>The lower half of the MIDI keyboard is mapped to pitched beeps of the construction vehicles. Upper half includes various “noisy” sounds.</td>
</tr>
<tr>
<td>Crickets</td>
<td>Cricket sounds recorded at night</td>
<td></td>
</tr>
<tr>
<td>Harp</td>
<td></td>
<td>Commercial sampled harp instrument from Native Instruments.</td>
</tr>
<tr>
<td>Clarinet</td>
<td></td>
<td>Another commercial instrument.</td>
</tr>
</tbody>
</table>

The score for *UM2012* is somewhat different from the score for the Demo Lecture version discussed previously. The biggest difference is that the Demo score provides three different state machines, one for each electronic instrument. It also specifies the instruments: harp, mourning dove, and construction. The *UM2012* score does not specify instrumentation. This leaves open the possibility of performances with or without the
electronics of the Motivator. A performance could be realized solely by the Motivator, or solely by live performers, or, ideally, by a combination of both.

In addition, *UM2012* only provides one state machine. All performers follow the same machine. However, the piece provides three different starting points. To quote the performance notes (see Appendix A):

To begin, each performer chooses a Starting Point from among the designated Starting Points. This indicates the first state for that performer. Performers may enter the performance at any time. Whenever the performer chooses to enter, he or she should simply choose a Starting Point and begin.

All of the state paths potentially overlap, so that if a performer begins at Starting Point 3, he or she may eventually end up in the same state as someone who begins at Starting Point 1. This approach allows for the potential of arrival points at which everyone is playing together (or at least from the same material) while providing numerous and unpredictable paths to reach those points.

This score also specifies minimum durations for some states and for some events. These durations are meant to be approximate, and are designed to prevent overly rapid changes from one state to the next.
Chapter 3
COMPOSITION AND THE PERFORMANCE SPACE

Audience Behavior and the Aural Environment of the Performance

In much of contemporary concert life, the performance space has, to a large extent, been standardized along with certain social mores that guide audience behavior during a performance. Certainly, there are many acoustic and spatial differences among concert halls, but certain features of the situation in a concert hall can usually be assumed. First, the musicians will be arranged on a stage separated from the audience. Second, the audience will be present before the piece begins, and will leave at some point after it ends. Audience members will remain for the entire piece. Third, the audience will avoid contributing to the aural experience of the composition by remaining silent throughout. Similarly, throughout the performance the concert hall will be silent except for the sounds that are intended to be part of the composition. Finally, although exceptions to the above features will certainly occur (for example, audience members talking, coughing, or coming in late; the sounds of the air conditioner or traffic outside), these exceptions are viewed as transgressions of the space and are either ignored or treated as nuisances. They are not considered integral to the piece, and the composer and performers are not expected to deal with them.

This situation is slightly different when a performance occurs outdoors in a public park or similar venue. In such a situation, it is unrealistic to expect silence, and the audience is likely to be somewhat less restrained in their movements. However, from the standpoint of the performance and the composition, the assumptions largely hold true. The space is often treated as a concert hall moved outside, with more exceptions to the
desired quietude. But these exceptions are still likely to be viewed as extraneous to the performance rather than an integral part of it.

This arrangement has proven very beneficial to the development of concert music. It allows composers to compose free of certain considerations. No one will criticize the composer for failing to take into account the frequencies of the air conditioner fan, or the experience of a listener who arrives halfway through the performance.

Table 3.1. Differences approaches to the sonic environment and audience behavior between the traditional concert hall and *Blue Skies*.

<table>
<thead>
<tr>
<th></th>
<th>Concert Hall</th>
<th><em>Blue Skies</em></th>
</tr>
</thead>
<tbody>
<tr>
<td>Performer-Audience</td>
<td>Separated</td>
<td>Intermingled</td>
</tr>
<tr>
<td>Audience Presence</td>
<td>Duration of piece</td>
<td>Unpredictable</td>
</tr>
<tr>
<td>Audience Sounds</td>
<td>Approaching silence</td>
<td>Unpredictable</td>
</tr>
<tr>
<td>Aural Environment</td>
<td>Approaching silence</td>
<td>Not silent at all</td>
</tr>
<tr>
<td>Relevance of Extraneous Sounds</td>
<td>Ignored/Not relevant</td>
<td>Relevant to overall sonic experience.</td>
</tr>
</tbody>
</table>

In *Blue Skies*, I make a different set of counter-assumptions. First, the musicians will be intermingled with the audience rather than separated from them. Second, audience members (and even musicians) will likely not be present for the entire performance. There is a high likelihood (true of many sound installations) that most listeners will arrive after the piece has started, and will leave before it begins. Third, the audience will make sounds, either deliberate or incidental, and will therefore contribute to the aural experience of everyone. There will not be any expectation of complete or even attempted silence. Similarly, the performance space will not be silent, and in fact will contain all of the usual sounds that occur in the area. Finally, these extra sounds and circumstances will
not be ignored, but rather will be considered part of the piece, and will thus be subject to further expansion of ideas within the piece (such as development as a motive, etc.) Table 3.1 summarizes the differences between these two approaches.

**John Cage and Max Neuhaus**

This approach to space has many important precedents. Perhaps the most iconic piece to transgress the standard model of concert hall performance is John Cage’s 4’33”, which Cage often simply referred to as his “silent piece.” Cage’s notion of “silence” is rooted in his distinction between “intentional” and “nonintentional” sound, with “silence” consisting of sounds of the unintentional variety, not carrying the baggage of some composer or performer’s decision-making process. In fact, Cage often expressed his preference for the latter. Cage viewed his piece as something that could be translated into just about any situation:

> I have spent many pleasant hours in the woods conducting performances of my silent piece, transcriptions, that is, for an audience of myself, since they were much longer than the popular length which I have had published. At one performance, I passed the first movement by attempting the identification of a mushroom which remained successfully unidentified. The second movement was extremely dramatic, beginning with the sounds of a buck and a doe leaping up to within ten feet of my rocky podium. The expressivity of this movement was not only dramatic but unusually sad from my point of view, for the animals were frightened simply because I was a human being. However, they left hesitatingly and fittingly within the structure of the work. The third movement was a return to the theme of the first, but with all those profound, so-well-known alterations of world feeling associated by German tradition with the A-B-A. ¹⁰

This quote reveals much about Cage’s approach. On the one hand, it displays his embrace of all sounds (and indeed all events) as worthy of attention. On the other hand, in his

---

witty style, Cage asserts himself as the artist in this situation: “they left hesitatingly within the structure of the work.” Or perhaps a better term is the “curator” of the situation. Cage is simply creating a three-movement temporal structure in which things happen and he pays attention to them. Blue Skies also embraces the presence of everyday sounds while simultaneously manipulating recordings of them. In a similar fashion to Cage’s woodland performance, the piece plays with the dichotomy between intentional and nonintentional sound: certain sounds that are simply part of the existing soundscape (and thus “nonintentional”) are incorporated into the piece and thus made “intentional”, asserting the composer’s authorial status.

Another composer who engaged even more directly with outside sounds and outside spaces was Max Neuhaus. In his series of works titled Listen, beginning in 1966, Neuhaus would direct audience members on a silent, guided walk through the urban environment of New York City:

Announced as concert programs, they were broadcast by word of mouth: participants were met by the artist at designated meeting points at prearranged times. After he had stamped the word LISTEN on their hands, Neuhaus took them on a walk that followed a carefully designated route, during which they encountered a series of soundscapes that included the heavy rumble generated by a Con Edison power plant, the vibrant street noise of a congested Puerto Rican neighborhood, and the roar of a freeway.11

These walks repositioned the audience as active participants in the “performance” of the composition. The composition in this case is a planned walk through various soundscapes. By walking a particular route, Neuhaus could predict to a large extent the sort of sounds that the audience would hear, while still allowing for the near certainty of

unplanned sounds also entering the mix. But since the audience members were the ones doing the walking, and since the walk essentially was the composition, the audience assumed a dual role of both listener and performer.

This notion contrasts radically with a more traditional concert hall model (and also that of the jazz club, the rock concert, the DJ set, etc.) For one thing, the performance of the piece shifts from the creation of the sound to the observation and appreciation of the sound. In Neuhaus’s sound walks, the workers at the Con Ed plant or the drivers on the freeway presumably were unaware of their participation in a piece of music. The locus of composition also shifts. We might reasonably define musical composition as the selection of sounds and the authoring of instructions to perform those sounds. But in *Listen*, Neuhaus’s instructions do not tell the performer what sounds to make. If the performers are the walkers, then the instructions instead create a situation in which the performers will simply be able to observe and appreciate certain sounds.

Although *Blue Skies* is a different sort of piece, it shares this goal of “creating a situation.” When we conceptualize composition as an act of establishing a situation in which performers, the audience, and other factors intertwine in certain ways, we can begin to think of ourselves as “designing” a composition the way one designs a car, a building, or a landscape, rather than “writing” a composition the way one writes a novel. These are not mutually exclusive perspectives, but the different viewpoints can be instructive. I will take up these ideas in more detail in the next chapter.
Chapter 4

EXPANSION OF INTERACTION THROUGH COMPOSITION

Any musical experience relies on the interaction among different entities. In the case of a symphony orchestra concert, we have at minimum the performing musicians, the conductor, the audience, the composer and the score, as well as the musical instruments and the concert hall space. Each of these factors contributes in some way to the overall musical experience. In the case of Neuhaus’s *Listen*, we have the composer/leader, the audience/performers, the urban space, and the various sound-producing entities.

In *Blue Skies*, we have a similar set of contributory factors. But a piece such as this has the opportunity to expand the possibilities for interaction beyond the traditional model. This chapter presents some of these possibilities.

Expansion of Interaction with the Environment

In *Blue Skies*, sensors detect variations in wind and light levels. This sensor data is fed into the algorithmic computerized improviser, which then alters the direction of its performance based on the input. Other types of sensors can also be introduced including those that respond to sound, temperature, humidity, and so forth. In addition, the score instructs performers to react in similar ways. This aspect of the piece gives it a distinctive marketing gimmick, but what are the aesthetic reasons for choosing to compose in this matter?

Clearly, the answer to this question is somewhat personal. My own involvement in this idea goes back several years, when I decided to write a piece in which the performers responded to environmental cues, choosing sections to jump to based on these cues. These cues might include any number of things, but would be motivated at least in
part by the location in which the piece was being performed. For example, if the piece was to be performed in an outdoor public urban space, cues might include, “You see someone wearing a red shirt,” or “The sun goes behind a cloud.” Although this piece was not formally realized, Example 4.1 shows an example of a similar piece, titled Reactions.

Example 4.1. Excerpt from Reactions, demonstrating composed responses to environmental cues.

This early piece demonstrates many of the concepts that inform my current work. First, there is the site-related form of the piece. Although this excerpt could certainly work in many different locations, it would be best suited to locations in which the various cues are likely to occur with enough frequency to keep the piece interesting. Second, there is the guided improvisation, with the performers given leeway to make decisions, but only within the parameters established by the environment as mediated through the score.
Aesthetically, the piece is also informed by the same motivations as Blue Skies. Most importantly, there is the desire to engage the performers in the experience of their environment. As with Cage, this approach is informed by my own studies of and engagement with Buddhist practice, in which attention to the present moment leads to a higher state of consciousness. By instructing performers to pay attention to their surroundings and use those surroundings to guide their improvisation, it is hoped that the performers’ overall engagement with the world will increase, even outside of the context of the performance.

This point addresses the importance of participation to the overall success of the piece. The composition is not intended only to create a satisfying musical experience, nor is it intended primarily for the enjoyment of the audience. Rather, it is intended to create a situation in which the performers interact and create in a particular manner. It may be instructive to think about composition in this way as a design discipline. The score in this paradigm becomes a mechanism for guiding the behavior of participants.

To illustrate this idea, consider a very simple building with four walls, a roof, and a single doorway, built on an empty lot somewhere that leaves ample space on each side of the building. A plan for this building is shown in Figure 4.1. Before the building is constructed, anyone who arrives at the empty lot is free to move about the space at will. But once the building arrives, the space has been divided into two distinct areas: inside and outside. However, it is not only the space that has been divided; the single doorway has now established a mechanism for controlling how visitors (or residents) move within the space. If someone wants to move from outside to inside, or inside to outside, they must pass through the doorway.
Figure 4.1. Behavioral guidance before and after the introduction of a building. On the left, the person is free to move about the central area of the empty lot. On the right, the person’s behavior is guided by the construction of the building.

The result is that human behavior has been guided and shaped. This shaping of behavior can be seen as one of the primary objectives of design. Consider another example: a standard dinner fork. The fork typically has a handle, and another end that contains the tines used to pick up food. The shape of the fork strongly directs our behavior in using it. For example, we could pick it up by the tines and attempt to stab a piece of food with the handle, but this would probably not work as well as the other way around. The design of the fork establishes certain parameters that will direct us to behave in certain ways.

Note that this behavioral direction need not be seen in a negative light as forced coercion. In fact, it may more often be a positive goal. Note as well that with our two examples, there are ways to circumvent the desired behavior. Certainly, we might find that the fork’s handle makes an excellent butter knife, and indeed it may largely be social convention that prevents us from using it in that manner. Similarly, we might be able to
take a sledgehammer and tear a hole in the wall of our building in order to bypass the designed behavior of the door, but we generally agree not to do that.

This same model can be applied to musical composition. We might think of a Beethoven piano sonata as a design that shapes the behavior of a pianist. There is, of course, no physical constraint that prevents the pianist from stepping outside of the prescribed notes and playing something else, but generally the classically trained performer has an unspoken agreement with his or her audience to stick closely to the boundaries established by the piece. In a fully notated composition, these boundaries are typically fairly strict, but in a less deterministic piece, the restrictions may be closer to those set aside by our one-room building.

In *Blue Skies*, we can start the design process by defining our participants and other contributory factors. Table 4.1 shows each of these factors and the role each one plays in the piece. First, we have the Environment, which provides cues to the performer and also provides the sonic environment in which the piece occurs. The Composer’s role is to select the site and decide the acceptable ways in which the performers, both human and electronic, should react to the environment. The Performers pay attention to their surroundings and make decisions about what sounds to produce based on their interpretation of the score. The Score then acts as a mediator among the composer, the environment, and the performer. The Listener might simply listen to the music, or might function as part of the Environment by providing cues, or might become a performer by making sounds. Finally, the Site itself informs the Composer’s decisions. Once we have identified these factors, the compositional process involves making decisions about how to restrict and guide the interactions of these factors.
Table 4.1. The role of each contributory factor in the piece.

<table>
<thead>
<tr>
<th>Contributory Factor</th>
<th>Role</th>
</tr>
</thead>
<tbody>
<tr>
<td>Environment</td>
<td>Provides cues to the performer</td>
</tr>
<tr>
<td>Composer</td>
<td>Indicates acceptable ways to respond to environmental cues, selects site, etc.</td>
</tr>
<tr>
<td>Score</td>
<td>Mediates between composer, environment, and performer.</td>
</tr>
<tr>
<td>Performer</td>
<td>Perceives cues, and makes decisions about sounds to produce based on the score.</td>
</tr>
<tr>
<td>Listener</td>
<td>Listens, or functions as part of the Environment (providing cues), or functions as a Performer.</td>
</tr>
<tr>
<td>Site</td>
<td>Informs Composer’s decisions about which cues will be available and should therefore be included in the score.</td>
</tr>
</tbody>
</table>

**Level of Human (Audience) Interaction**

Human interaction with the piece is an important consideration. We can consider three different approaches to this aspect of interactivity. First, there is intentional interaction, in which the listener deliberately manipulates the composition. This level of interaction can be achieved by providing an interface that responds in a very direct way to the interaction. This gives the participant the sense that he or she is directly causing an effect in the piece. Most (if not all) traditional musical instruments work this way: striking a key on the piano causes a note to sound. Similar interfaces can be developed for interaction with a computerized sound generator. For example, proximity sensors can be used to alter sound according to how close a person is to particular locations in the space. Or, cameras
can be used to track a person’s location and manipulate sound accordingly. An interesting example of this is installed in the observation deck of Rockefeller Center in New York City, known as “Top of the Rock.” In the Target Interactive Breezeway, created by Cameron McNall and Damon Seeley of the group Electroland, colored lights in the ceiling track an individual upon entering the room. The software driving the system keeps track of each person, assigning a particular color to that person. Entering the room, then, someone is able to “play” with the artwork, causing it to respond.\textsuperscript{12}

A related type of interaction is interaction without intent. In this style, the system responds to user behavior, but in a manner that obfuscates a direct causal link to the user. This sort of interaction can be realized with the same sort of technology used in intentional interaction, but with a different response. For example, a system might shift to a different state based on the number of people that are in a room. A count could be taken using video tracking or some other means. When the count is below ten, for example, the piece would be in state A, but when the count exceeds ten, it would transition to state B. In this way, people who come to experience the artwork are affecting its behavior, but they may not even be aware of it. Furthermore, it will be more difficult for an individual to directly alter the behavior of the piece, as the response is based on collective behavior.

This last point is an important consideration when designing a particular installation. To what degree should observers feel that they have some control over the direction of the piece? There is no right or wrong answer; it is simply a point of consideration.

The third and final model of user interaction is to simply avoid it altogether. In this case, observers don’t have any direct effect on the piece. Note that these three models are not mutually exclusive within a particular piece. For example, there may be some programmed behaviors that respond to the audience in an obvious way, while others respond in more subtle ways that may not be apparent to participants. Or, certain interactions may be enabled only in response to certain other environmental events, so that sometimes the piece interacts with the user, while at other times it does not.

**Complexity of Interactions**

A related point to consider during the composition of an interactive soundscape composition is the complexity of interactions within the piece. These interactions might be with people or with other of aspects of the external environment, such as weather. Generally speaking, it seems safe to postulate that the following features would make interactions more complex: more inputs, more states for the system, and more parameters that are affected by the inputs. Complexity is relevant to the overall perception of the piece because interactions that are more complex may be less likely to be perceived by the audience as interactions at all. For example, if a piece tracks 10 different inputs and has 50 parameters that are affected by the inputs in different combinations, it seems likely that an audience member will have a harder time discerning these relationships than a scenario in which one input affects one parameter.

Again, there is no right or wrong approach; this is simply an aesthetic consideration. A simple network of relationships allows observers either to “play” the composition (in the case of human-computer interactions) or to notice clear correlations between environmental changes and the development of the music. However, a complex
network might bring the piece closer to Cage’s ideal of “nonintentional” composition. If complexity reaches a point where the relationships become difficult to perceive, then the piece might give the illusion of being randomly motivated.

Sometimes, however, this sort of interaction can lead to musically exciting results that surpass the intention of the individual performers. As Gil Weinberg points out, these performance systems form a network:

The shape of the composition in such systems grows from the topology of the network and its interconnections with the performers. Such an environment that responds to input from individuals in a reciprocal loop can be likened to a musical “ecosystem.” In this metaphor, the network serves as a habitat that supports its inhabitants (players) through a topology of interconnections and mutual responses that can, when successful, lead to new breeds of musical life forms.\(^\text{13}\)

Careful consideration of this network topology can help the composer better understand the sort of interactions that a composition will promote. Weinberg’s work aims at a comprehensive theoretical description of these mechanisms. Although Weinberg is specifically describing “live performance systems that allow players to influence, share, and shape each other’s music in real-time,” it may still be beneficial to apply certain aspects of his approach to Blue Skies. Weinberg positions two paradigms at opposite poles of a range of musical networks. On one end are “process-centered” musical networks, in which “the musical outcome of the interaction is usually less important than the process participants undertake while creating this outcome.” At the other end are “structure-based” systems, in which the “main goal of the interaction tends to focus on its

---

outcome, whether it is the music or the performance.” Most networks will contain some elements of both.\(^{14}\)

*Blue Skies* would seem to fall somewhere between these two extremes. An important aspect of the piece is the experience of the performers and their creative engagement with the other musicians within the musical constraints presented by both the score and the environmental events. However, certain musical goals are desired, especially relating to the choices of material used in the computer-generated portion of the piece. By drawing raw source material from the site where the piece is to be performed, it is hoped that listeners will gain a greater awareness of these sounds as they occur outside of the piece. This aspect falls under the category of the “artistic vision” of the composer that is such an important part of structure-based systems, even if most of the piece would be best described in process-oriented terms.

Chapter 5

REFLECTIONS AND FUTURE WORK

Rather than a finished composition, *Blue Skies* provides a blueprint and a platform for future work. There are many opportunities to improve upon the platform, and many opportunities to expand the artistic scope of the piece. Any site can serve as an interesting source of material, and it would be enlightening to install and perform the piece in a series of different locations.

There are several technical considerations for moving forward with the piece. In its current format, the platform uses a limited range of sensors. A logical next step is to expand the scope of this range to detect other types of data and explore alternate ways in which the sensors can be used to control the sound. In addition, the human interaction with the piece is fairly limited. Further opportunities exist to explore the different degrees of human interaction discussed in Chapter 4.

Another interesting idea is to port the Motivator software to mobile devices such as the iPhone or Android. An exciting potential exists to create a “product” that users could download to their mobile device. This could be combined with a small hardware weather station that would track weather conditions wherever the user is and adjust the music accordingly. New compositions could be downloaded from a web site. The user could either listen to the music on headphones, or plug it into a speaker system and play it at their home or office. The application or web site might include a user interface that would allow other people to compose their own compositions for this platform.

Another approach that merits further exploration would be to create small, modular hardware components, each consisting of a computer (such as an iPhone or similar device), a speaker (or a couple of speakers) and a wireless receiver to receive
sensor data. This is similar to the model used in the current system (discussed in chapter 2 and shown in Figure 2.3), but instead of having a single, central computer responding to the sensor input and handling all of the algorithmic improvisation, each unit would respond separately to input and produce its own improvisations based on its own set of sounds. This approach would physically reflect the autonomous nature of the different electronic “performers,” corresponding more explicitly to the autonomous nature of human performers. If the cost-per-unit could be kept low enough, this approach could allow for easily scaling the system from two or three speaker units to dozens, potentially spread about in different locations throughout a site, without any need for them to be physically connected. Both of these directions would provide new design opportunities.

Earlier in the paper I asserted that it was best to consider Blue Skies as a template for producing multiple instances tailored to different sites and different circumstances. In some ways, the issue of whether Blue Skies is a “composition,” a “process,” or something else entirely is an unresolved question. The difficulty of reaching a definite conclusion may reflect an imperfect fit between the terminology and this sort of art. George Lewis struggles with this question as well with regard to his improvising computer system Voyager:

Voyager's unusual amalgamation of improvisation, indeterminacy, empathy and the logical, utterly systematic structure of the computer program is described throughout this article not only as an environment, but as a "program," a "system" and a "composition," in the musical sense of that term. In fact, the work can take on aspects of all of these terms simultaneously—considering the conceptual level, the process of creating the software and the real-time, real-world encounter with the work as performer or listener. Flowing across these seemingly rigid conceptual
boundaries encourages both improvisors and listeners to recognize the inherent instability of such taxonomies.\textsuperscript{15}

This observation highlights the difficulties of classifying music that does not obviously conform to the standard compositional paradigm. The question is not merely one of semantics, as it can have profound implications in the development of music school curricula, grant programs, and performing arts institutions.

A related and potentially problematic aspect of a composition like \textit{Blue Skies} is the assessment of the “success” of the piece. Composers often speak of whether or not a composition is “successful.” One feature of a composition like \textit{Blue Skies} is that in many ways, the actual \textit{sounds} of the piece are secondary to the \textit{concept} of the piece. Ultimately, the aesthetic satisfaction will arise from the experience of particular sounds and from the experience of engaging with the piece and with the other participants. But the composer has to concede a great deal of control over what these sounds are. Thus, in assessing “success,” we should perhaps return to the notion of the piece creating a situation in which sonic events happen, and a situation in which performers are guided to behave musically in certain ways. For \textit{Blue Skies}, we might then examine questions such as the following: Does the piece create a situation that is satisfying to the participants? Do the participants finish the piece with a higher awareness of their surroundings than they had before? How well does the piece integrate with the existing soundscape? Does it create a good balance between blending with the soundscape and asserting its presence?

We might also consider technical questions, such as: Are the different sounds well recorded and edited? Does the music respond effectively to events in the surroundings? Are the relationships between environmental events (such as the wind) and the sounds

\textsuperscript{15} Lewis, “Too Many Notes,” 33.
produced by the piece convincing and clear? Are the mechanics of the piece transportable to other situations? Does it work continuously? In assessing the use of live performers, we might ask: Do the performers have a clear idea of what to do? Does the piece create a situation that allows the performers to engage with each other and their surroundings in interesting ways? Does the piece provide a space for improvising performers to do something that they wouldn’t do without the piece?

All of these questions are worth pursuing, and all of them provide fertile ground for further discussion of the Blue Skies template and any individual compositions that it spawns. It is my hope to develop many more pieces along these lines as I continue to explore these ideas.
BIBLIOGRAPHY


Keller, Damián and Ariadna Capasso. “New Concepts and Techniques in Eco-


Rowan, Dean C. "Modes and manifestations of improvisation in urban planning, design, and theory." *Critical studies in improvisation/Études critiques en improvisation* 1, no. 1 (January 1, 2004).


APPENDIX A: SCORE FOR UM2012 EDITION

The score for Blue Skies: UM2012 is formatted into three pages, with an additional page for front matter and instructions. All performers play from the score. The three pages are to be lined up side-by-side to construct a single large diagram. The following pages contain this score for study and reference purposes.
Blue Skies
Dan Dickinson

This piece is written for any instrumentation and any number of performers. It is designed in conjunction with the Blue Skies computer system, a hardware/software system that includes sensors to detect changes in wind and light level, and an algorithmic improviser that produces sounds in response to these changes. The score can be played both by people and by the Motivator. Translation into the Motivator’s file format is required in order for it to play the score.

The piece should be performed outside.

Each rectangle represents a State. When you enter a state, follow the instructions in that state.

Arrows emerging from states indicate transitions. Most arrows are labeled either with verbal instructions or with a symbol (explained under “Special Symbols” to the right). This symbol or instruction indicates the trigger that should cause you to transition out of your current state. If none of the triggers occur, you should remain in your current state until one occurs. Most of the triggers ask you to pay attention to either the wind or light level in the performance area and respond accordingly.

To begin, each performer chooses a Starting Point from among the designated Starting Points. This indicates the first state for that performer.

Performers may enter the performance at any time. Whenever the performer chooses to enter, he or she should simply choose a Starting Point and begin.

Any musical parameter that is not specified is to be determined by the performer.

If a staff is shown without a clef, then any pitches are determined by the performer. However, within the performance of a particular figure, the performer should be consistent about the pitch that is tied to each line or space. The interval between two adjacent lines or spaces should still be roughly a 3rd (major or minor).

A symbol followed by an amount of time indicates that you should not transition until that condition has been maintained for that amount of time. For example, the “little or no wind” symbol followed by 30” would indicate that you should transition once the wind has stopped (or almost stopped) for 30 seconds. These times should be viewed as approximate; it is not necessary to use a stopwatch unless you want to.

A time without a symbol indicates that you should transition after being in the state for the specified amount of time. Again, the time is approximate. Use your best judgment!

The performance continues indefinitely. Whenever you wish to stop playing, simply stop.

For more info, contact the composer:
Dan Dickinson
dp dickinson@yahoo.com

© 2012 Daniel P. Dickinson, BMI.
Starting Point 1

Play one of the following motives:

Pause, then play the same one or choose a different one. Repeat for one minute then move to the next state.

Blue Skies
Dan Dickinson

Big Gesture

Play this:

repeat indefinitely

Aeolian Pitch

Observe the wind. If there is no wind, play this at pitch. If the wind picks up speed, play this proportionally higher based on the speed of the wind.

more wind
higher pitch

You decide what "proportionally higher" means.

Notes in parentheses are optional.

Time between notes is roughly proportional to horizontal space.

lower
less wind

Improvise 5th

Improvise using these notes:

Maintain a moderate tempo.

Massive Chord

Play this once:

(Pick any note and hold it.)
(Pick another note and hold it. If possible, keep holding the first.)

(Pick another note and hold it. If possible, keep holding the first.)

(Pick another note and hold it.)

Do nothing.

Nada Dot

Do nothing.

Empty

Do nothing.

Page 1
Starting Point 2

Nada

Do nothing for one minute.

Choose one of the other performers* to follow.

Imitate the rhythm of whatever that performer plays,
just a little behind them in time.

For the pitch:
Choose a pitch to be the “center”.
If it’s dark, just play that pitch.
If it’s brighter, alternate between two pitches
an equal interval above and below the center pitch.
As it gets brighter, increase the interval.
As it gets darker, decrease the interval.

* Performer may be a person or one of the
computer-controlled instruments.

Starting Point 3

Bursts of Noise

Play bursts of noise.

Improvise 2 short motives with the following scale:

Continue to improvise using only the same 2 motives.
Pause 3 or 4 seconds after each motive.
You may subtly alter the motives each time.
If you have already been to this state, reuse your motives!

Sustained Noise

Play sustained noise. Get louder as the wind gets stronger.

Choose one of the other performers* to follow.

Imitate whatever that performer plays,
just a little behind them in time.

* Performer may be a person or one of the
computer-controlled instruments.

Improv 5th Rit

when it gets really slow

Start fast.
Gradually get slower and slower.
**Wind Popcorn**  
Play short percussive sounds.
- If there is more wind, increase the density of sounds.  
- If there is less wind, decrease the density.

Use a variety of sounds, but keep them short!

**Slow Growing Chord**  
- Play this over and over:
  - Then this after a while:
  
As the performance space gets darker, *accelerando.*  
As it gets lighter, *ritardando.*

**Crickets**  
Choose a high note.  
Play in short, staccato pairs.

Rest between pairs is shorter when space is darker, longer when it's brighter.

**Beep, Beep, Beep.**  
Choose one pitch.  
- Pick a tamp, and play your pitch in a rock-solid rhythm:
  - Mechanically
  - Repeat indefinitely.

You may gradually vary dynamics, but do not vary pitch, rhythm, or accent.

**Empty 2**  
Do nothing.

**Sporty Canon**  
Choose one of the other performers* to follow.
- Occasionally, *imitate* one or two notes that performer plays, just a little behind them in time.
  * Performer may be a person or one of the computer-controlled instruments.

**Canon 3rd**  
Choose one of the other performers* to follow.
  * Performer may be a person or one of the computer-controlled instruments.
APPENDIX B: SCORE FOR DEMO EDITION

The Demo Edition of *Blue Skies*, for harp, mourning dove, and construction sounds, is presented in this appendix. The entire score is formatted for one large page, and is presented first. To aid readability, the part for each instrument is then repeated on its own subsequent page.
Compose 3 motives using this Just-Intoned scale:

Start Here!

Day

Pick one of the motives and play it.
Try new ideas and play.
Keep doing this.

Day 2

Pick one of the motives and play it.
Try new ideas and play.
Keep doing this.

Night

Pick one of the motives and play it.
Try new ideas and play.
Keep doing this.

Night 2

Pick one of the motives and play it.
Try new ideas and play.
Keep doing this.

Canon 2

Pick whatever the harp plays, but a Major 3rd higher.
Pick whatever the harp plays, but a perfect 4th higher than the original, and holding each note 3 times as long.
Pause for 3 or 4 seconds, then pick another one and play it the same way.
Keep doing this.

Canon

Pick whatever the harp plays, a Major 3rd higher.
Play whatever the harp plays over and over.

Construction

Play construction sounds over and over.

Beeps

Play whatever the harp plays, a Major 3rd higher.
Play whatever the harp plays, a Major 3rd higher.

Noisy

Play whatever the harp plays.

Wait

Do nothing.

Too fast!

Wait for Day

Do nothing.

Too slow!

Demo Composition for Indoor Classroom (Volpe 103)

Blue Skies

Dan Dickinson
Compose 3 motives using this Just-Intoned scale:

Start

Composing 3 motives using this Just-Intoned scale:

Day
Pick one of the motives you composed and play it
Then pick another one and play
Keep doing that.

Day 2
Pick one of the motives you composed and play it
Then pick another one and play. Keep doing that.
Speed up and slow down your tempo with the wind.

Night
Improvise using these notes:

Night 2
Improvise using these notes:
Gradually get slower and slower.

Wait for Day
Do nothing.

Start Here!

Demo Composition for Indoor Classroom (Volpe 103)
Compose 3 motives using this Just-Intoned scale:

Start Here!

Day
Pick one of the motives you composed and play it. Then pick another one and play. Keep doing that.

Night
Pick one of the motives you composed and play it. But a perfect 4th higher and holding each note 3 times as long. Keep doing that.

Canon
Play whatever the harp plays, a Major 3rd higher.

Canon 2
Play whatever the harp plays (same pitch) but hold each note a little bit longer.

Start
Compose 3 motives using this Just-Intoned scale.
Compose 3 motives using this Just-Intoned scale:

Start Here!

**Wait**
Do as little as possible.

**Noisy**
Play construction sounds over and over.

**Beeps**
Play only the beeps of a truck backing up, gradually getting faster.

**Construction**

Very Strong Wind!

Wind getting weaker...

Start Here!

Too fast!

Demo Composition for Indoor Classroom (Volpe 103)
APPENDIX C: MAX PATCHES

This appendix presents the implementation of the Motivator in the Max software. Rather than present every single Max patch that was developed for this project, we will examine a few representative patches. In addition, it should be noted that the Motivator project is still in development, so that the images below are properly viewed as a snapshot of the current state of the project as it exists at the time of this writing rather than a final, definitive version. These patches make heavy use of the dictionary data structure introduced to the product in Max version 6. A dictionary is simply a lookup table that allows the developer to associate keys with values. The value can then be discovered by looking up the key. For example, in a phonebook, the keys would be names, and the values would be phone numbers. This is the basic model for a dictionary data type.

Frustrated by the spaghetti mess of patch chords that greeted me every time I made a change to the existing patch or attempted to fix a bug in early versions of the Motivator, I developed a framework that I named “oop”, after the standard acronym for “object-oriented programming.” In this framework, each patch contains a dictionary that allows us to assign “properties” to an object. The framework also implements a reusable patch called “oop.method” that allows us to define “methods” or “functions” on our objects. This approach allows the Max developer to develop in a way that is familiar to users of object-oriented programming languages such as C++, Java, and Smalltalk.

Figure 12.1 shows an overview of the main Max patch for Blue Skies. This patch serves as a representative example of what any Blue Skies instance will contain. The patch contains several zul.Motivator patches, one for each instrument. The zul.Motivator patch implements the Composer/Improviser component discussed in Chapter 2. This main patch also contains a zul.StateMachine patch that corresponds to each instrument.
Together, the zul.Motivator and zul.StateMachine pair act as the “performer” for that instrument, interpreting the score and producing sounds. The patch also contains a zul.Sensors object that reads data from the serial port and converts it into the raw sensor messages described in Chapter 2. The patch also contains three zul.DataMonitor patches, one for each sensor type. (In addition to light level and wind speed, this patch also supports wind direction.) It also contains a zul.Clock object, which is capable of sending messages that indicate the current time. At the bottom of the patch are four zul.StateMachineUI patches that provide a user interface for loading score files into a state machine, choosing a current state, and displaying the current state.

Figure 12.1. Main Max patch for Blue Skies: UM2012

Figure 12.2 shows the main window of the zul.Motivator patch. On the left is an “oop.define” object. This is part of the “oop” framework discussed above. Inserting this
into a patch gives the patch basic object-oriented functionality. Below the oop.define object is a comment listing the properties that are available on the zul.Motivator. These correspond to keys in the dictionary provided by the oop.define object. Some of the interesting properties include pitch-ratio, which allows us to enter a tonic frequency and a list of ratios. This creates a list of pitches which the zul.Motivator will use to compose or improvise new material. The time, velocity, and duration properties similarly take lists of allowed values that the zul.Motivator can use in generating music.

Figure 12.2. zul.Motivator Max patch, which implements the Composer/Improviser component of the Motivator.

On the right side of the patch are quite a few subpatches, each of which implements a method or function presented by zul.Motivator. Figure 12.3 shows one of these method implementations, the play method. This method implements the basic functionality of the Composer/Improviser, which is to play notes. The method also demonstrates several components of the oop framework.
At the top of the patch is an oop.method object. This defines an object named “play” with a unique prefix generated by the Max system. (The “#0” prefix will be replaced at runtime with the unique identifier of the object that contains the patch.) Inside the oop.method patch is a Max receive object that has the “#.play” name. This allows us to send messages to this method. Near the bottom of the patch, we see several oop.call objects. These send a message to the oop.method object with the specified name. So, in our zul.Motivator patch, we have defined an oop.method called “#0.getNote” and one called “#0.playNote.” The oop framework makes it very easy to abstract this functionality.

Figure 12.3. Implementation of the play method in zul.Motivator.
Also demonstrated here is the *oop* object, which allows us to get and set properties on the object. The *oop* object near the top looks up the *playing* property by passing a “bang” message into the left inlet. This property will either have a value of 0 (if it’s not currently playing) or 1 (if it is currently playing). The property is set by passing a value into the right inlet. Underneath the hood, the *oop* object looks up the “playing” key in its dictionary and passes the value out of its outlet. Figure 12.4 shows the implementation of the *oop* patch. Subsequent figures show the *oop.method*, *oop.call*, and *oop.define* patches.

Figure 12.4. The *oop* object, which allows for easy setting and retrieval of object properties.
Figure 12.5. The oop.method patch. This patch allows for the definition of methods or functions similar to those allowed in more traditional programming languages.
Figure 12.6. The oop.call patch. This patch allows for the “calling” of an oop.method in a manner similar to a function call in a more traditional programming language.
Figure 12.7. The oop.define patch. This patch provide functionality that can be reused by all objects that use the oop framework.