Jason Freeman
Music Department
College of Architecture
Georgia Institute of Technology
840 McMillan Street
Atlanta, Georgia 30332-0456 USA
jason.freeman@music.gatech.edu

Extreme Sight-Reading, Mediated Expression, and Audience Participation: Real-Time Music Notation in Live Performance

Rossini would often wait until the last possible moment to compose overtures. He wrote the overture to *Otello* the evening before the opera's 1816 premiere. For *La gazza ladra* the following year, Rossini waited until the day of the premiere to score the overture, working, as he later wrote, "up under the roof of La Scala in Milan." He also noted: "Nothing is better for inspiration than necessity, the presence of a copyist waiting for your work, sheet by sheet" (Hughes 1956, p. 247).

Real-time music notation systems take Rossini's strategy to new extremes, waiting to create the score until during the performance. Unlike most live computer-music performance environments, these software algorithms do not produce digital audio or control data. Instead, they produce a dynamic musical score that may contain conventional Western notation or a range of graphical representations, which is interpreted by human musicians to create sound.

Such systems have benefited tremendously from recent technological advances. Lightweight, affordable devices such as laptop computers, LCD monitors, handheld PDAs, mobile phones, and video projectors display high-resolution graphics to be viewed by performers and/or audience members during a performance. These distributed devices can communicate with each other to build interactive performance systems using high-bandwidth, low-latency wired and wireless network protocols. Developers can create real-time video processing and animation software for powerful graphics cards using intuitive development environments. Furthermore, with the recent appearance of digital music stands marketed to replace printed scores, many musicians are becoming more comfortable with the idea of reading their music from a digital display. These advances have led to a corresponding growth over the last decade in the number of composers who develop real-time notation systems for their work.

In this article, I place real-time music notation within the context of algorithmic and computer-assisted composition and also within the aesthetic framework of open-form composition. I then discuss two key motivations for the pursuit of real-time music notation systems: mediating the output of live performance algorithms through human musical expression, and connecting audiences to musicians by allowing them to influence the behavior of those algorithms. Finally, I explore two major challenges associated with real-time notation systems—rehearsing the music and illustrating the process to the audience—and consider a variety of ways to address these issues. When composers creatively take advantage of the opportunities offered by real-time notation and address its design challenges, these systems can help them to create performance paradigms that redistribute the roles of composer, performer, and listener, substituting the unidirectional flow from composer's score to performer's sound to listener's ears with a collaborative feedback loop. The musical score is not only collectively interpreted; it is also collectively created.

Background and Context

Real-time music notation systems draw from a broad spectrum of algorithmic composition environments that produce music notation. They are also influenced by an open-form aesthetic in which a musical score is read differently in each performance of a composition.

Computer Music Journal, 32:3, pp. 25–41, Fall 2008

Computer-Assisted Composition

In the *Illiac Suite* (Hiller and Isaacson 1957) and other works, Lejaren Hiller and Leonard Isaacson transcribed the textual symbols produced by their software into musical scores for instrumental performers. These experiments in computer-composed music (Hiller 1981) resulted in printed notation, not electronic sound. The composers' reason for using notation was a practical one (they had no digital-to-analog audio converter), yet their experiments still serve as one of the earliest examples of music algorithmically generated by a digital computer (Hiller and Isaacson 1959).

Such notation-output systems have remained an important thread in algorithmic composition practice. For example, Gottfried Koenig's Project 1 environment (Laske 1981; Ames 1987), inspired by serial composition techniques, generated textual score tables that the composer could then transcribe into more traditional notation. Roberto Morales-Manzanares's Escamol system (Morales-Manzanares 1992) used non-deterministic automatons to generate scores in a variety of formats, including MIDI. Clarence Barlow's MIDIDESK software (Roads 1996) algorithmically manipulated existing MIDI files through techniques such as Markov-chain analysis and imitation. (Resulting MIDI files from both the Barlow and Morales-Manzanares systems could then be transcribed into musical scores.) David Cope's Experiments in Musical Intelligence (EMI) system used recombinant strategies to create new musical scores that are stylistically similar to those in a database of analyzed works (Cope 1996).

Such computer software often generates an entire musical score in a single execution; composers can also incorporate algorithmic software as one element within a broader compositional process. For example, Iannis Xenakis used his Free Stochastic Music program (Xenakis 1992) to generate material that he reworked and incorporated into several compositions, including *ST/10-1,080262* (1962). Current software programs supporting algorithmic score generation range from software libraries for major programming languages to markup languages and plug-in architectures for popular engraving applications to hybrid environments, such as PWGL's Expressive Notation Package (ENP), which combines graphical editing with scripting support (Kuuskankare and Laurson 2006).

In 1992, IRCAM created a research group for computer-assisted composition (CAC); the group developed the PatchWork and OpenMusic environments that have been used by composers such as Gérard Grisey, Tristan Murail, and Brian Ferneyhough (Assayag et al. 1999). Gérard Assayag sums up the philosophy behind many CAC tools:

> CAC provides the experimental environment that makes it possible to subject a very great number of musical instances resulting from a formalism to the test of musical quality. More, it authorizes the experimentation on formalisms themselves, which can be tested and in return modified or given up if they do not fulfill their promises (Assayag 1998).

CAC environments establish a relevant model for real-time music notation systems. In a CAC system, the composer manipulates the notation generated by the software to create a musical score by selecting, modifying, extending, and combining segments of the output. In a real-time notation system, the performers interpret the notation generated by the software to create the sound of each live performance.

Most notation tools for CAC environments, however, are difficult to incorporate directly into real-time notation systems, either because they write files directly to disk or because they must be controlled through graphical interfaces that are difficult to use in real-time performance. Nick Didkovsky's Java Music Specification Language (JMSL), in contrast, is well suited to real-time notation, because it complements its graphical JScore application with a library accessible both from Java and from Cycling '74's Max/MSP. The library enables programmers to algorithmically create and manipulate score data and to flexibly render those scores to a display or a graphics buffer with precise timing control; extended notation can then be added using Java graphics primitives (Didkovsky and Burk 2001).

Many other composers have developed their own real-time rendering software, using graphics primitives available in computer-music languages or writing in more general-purpose languages, in order to customize their environment to the unique visual language of a particular composition's notation (e.g., Winkler 2004; Kim-Boyle 2005) or to run the software on specialized display devices (e.g., Gutknecht, Clay, and Frey 2005).

Open-Form Composition

At the same time that Hiller and Isaacson were writing software for the ILLIAC, composers such as Earle Brown were rethinking the roles of the composer and performer in their music. The works of sculptor Alexander Calder, in which "the construction of units and their placement in a flexible situation . . . subjects the original relationships to constant and virtually unpredictable, but inherent, change" (Brown 1999, p. 40), inspired Brown to focus on the use of malleable notation in performance. He used the term *open form* to describe his scores in which the order of materials, and even their contents, could change dramatically from one performance to the next. In many of Brown's works, such as *Available Forms I* (1961), a conductor determines the order of the fragments and cues the musicians accordingly. In another example of an open-form work, Karlheinz Stockhausen's *Klavierstück XI* (1957), musical fragments on the printed page are played in the order in which the pianist's eyes wander across them. And composers such as David Kim-Boyle (2006) and Kevin Baird (2005) have acknowledged the influence of these open-form works on their real-time notation systems.

Brown's *Calder Piece* (1966) is an even more direct antecedent to real-time notation. In this work, a freestanding mobile sculpture, created by Calder specifically for the composition, sits among the four percussionists and serves as their conductor. During sections of the piece, the musicians are instructed to "read [the] moving mobile" (Brown 1966, p. 2), using the positions of elements of the mobile to determine which sections of the printed musical score (see Figure 1) to play (Rothstein 1981). The sculpture also serves as a musical instrument: players strike it at various times during the performance.

In *Calder Piece*, the open form jumps off the printed page into the real world: The physical traversal of the performance space by the mobile drives the visual traversal of the score by wandering eyes. Art Clay's 2005 composition *GoingPublik*, for trombone trio, similarly extends open form into physical space, using new technology and real-time notation to invite the performers to navigate an open-form score (see Figure 2) by exploring the performance venue. The nuances of their movements, detected by GPS units and three-dimensional compass sensors, select and transform the notation fragments they view (Gutknecht, Clay, and Frey 2005). The graphical notation—a combination of preloaded images, real-time image distortions and magnifications, a superimposed grid, and iconic and textual instructions—guides the players' musical improvisation and physical movement as they read it from head-mounted displays.

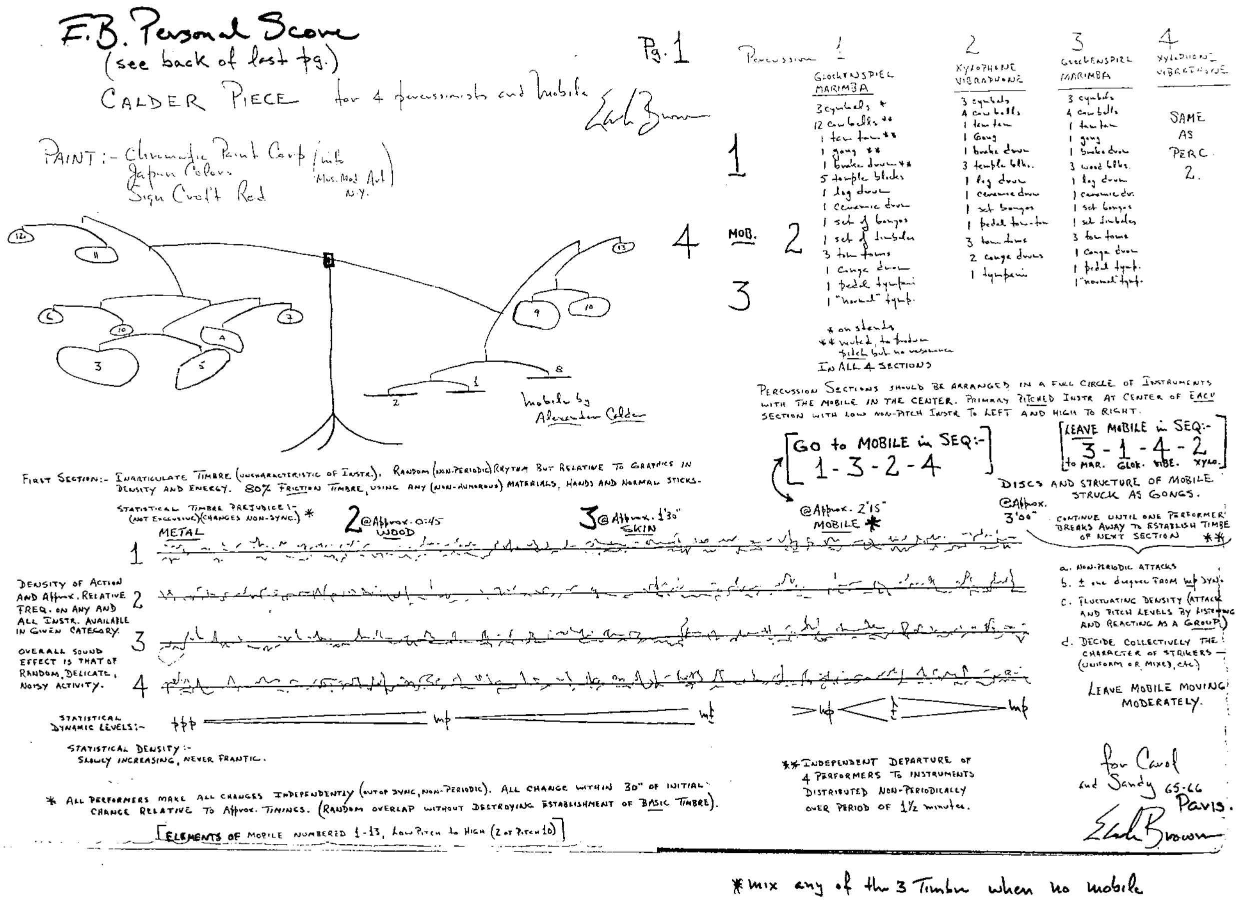
New Opportunities to Connect Performers and Audiences to Algorithms

By delaying the creation of music notation until the moment of performance, real-time notation systems create new opportunities for integrating human musicians with live performance algorithms and for connecting audiences to performers through those algorithms. The dynamic musical score becomes an expression of the algorithm and the process or people that drive it.

Merging Algorithmic Output and Human Musical Expression

Human musical expression plays an important role in most live performances of computer music, as the audio or control data created by performers of conventional musical instruments or new musical interfaces influence algorithms generating the computer's musical output. The software typically generates its output directly as digital audio or as control data for other music-making devices such as MIDI synthesizers or mechanical automatons.

Figure 1. Excerpt from Earle Brown, Calder Piece *(1966) for four percussionists and mobile (page 1). Courtesy of the Earle Brown Music Foundation.*



The performers respond to the computer-generated music to complete a feedback loop. The audience usually hears music originating from two sources: the algorithm and the musician(s) (see Figure 3a).

In real-time notation systems, not only can humans create input that drives an algorithm; they also musically interpret the algorithm's output into sound. The audience hears music originating from a single, merged source: the algorithm's output as interpreted by the musician(s) (see Figure 3b).

Musicians, of course, routinely interpret notation in concert to perform a musical work. But in most performances combining real-time algorithms with conventional instruments, musicians either play from a printed score, conceived in advance by the composer to complement any sounds the algorithm might produce, or they improvise without a score, interacting with the algorithm as an autonomous machine musician. Real-time notation systems, in contrast, merge the algorithm's output and the musician's interpretation of that output, creating a symbiosis of human and computer musical expression. Guy Garnett notes that such an elimination of this "dichotomy" between machine and human musicianship, "join[ing] the mechanical power of the machine to the 'subjectivizing' control of the human performer... is itself an aesthetic value for our new age" (Garnett 2001, p. 32).

Karlheinz Essl's *Champ d'Action* (1998) offers a simple illustration of this merged expression. Players in a chamber ensemble improvise, guided by notation (see Figure 4) that they read from laptop computer screens. The notation includes both graphical and textual elements with a set of symbols

Figure 2. Art Clay, Going Publik (2005) for trombone trio, excerpt from real-time music notation.

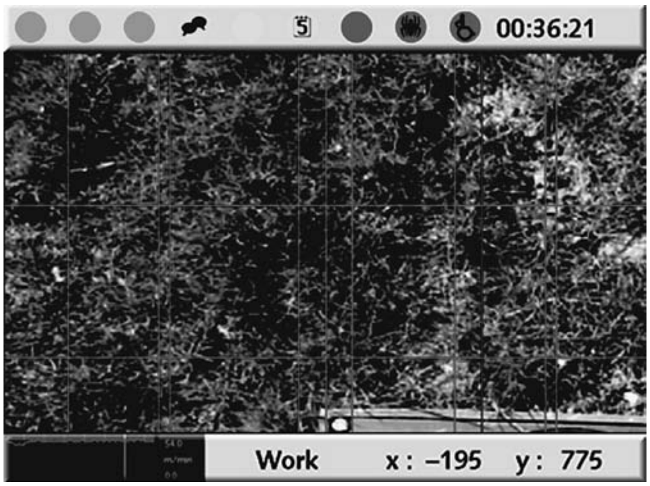
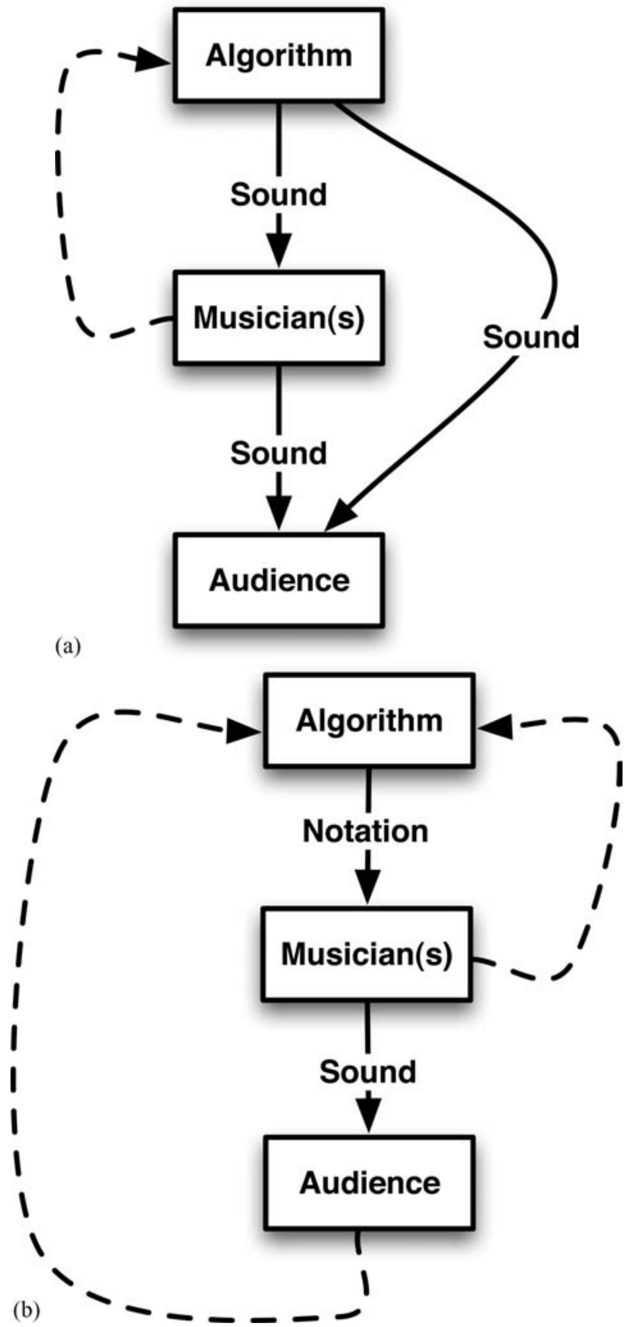


Figure 3. Typical connections among the output of an algorithm, musician(s), and audience when algorithms output (a) sound and (b) notation. Dotted lines indicate common input paths back into the algorithm to create an interactive feedback loop.



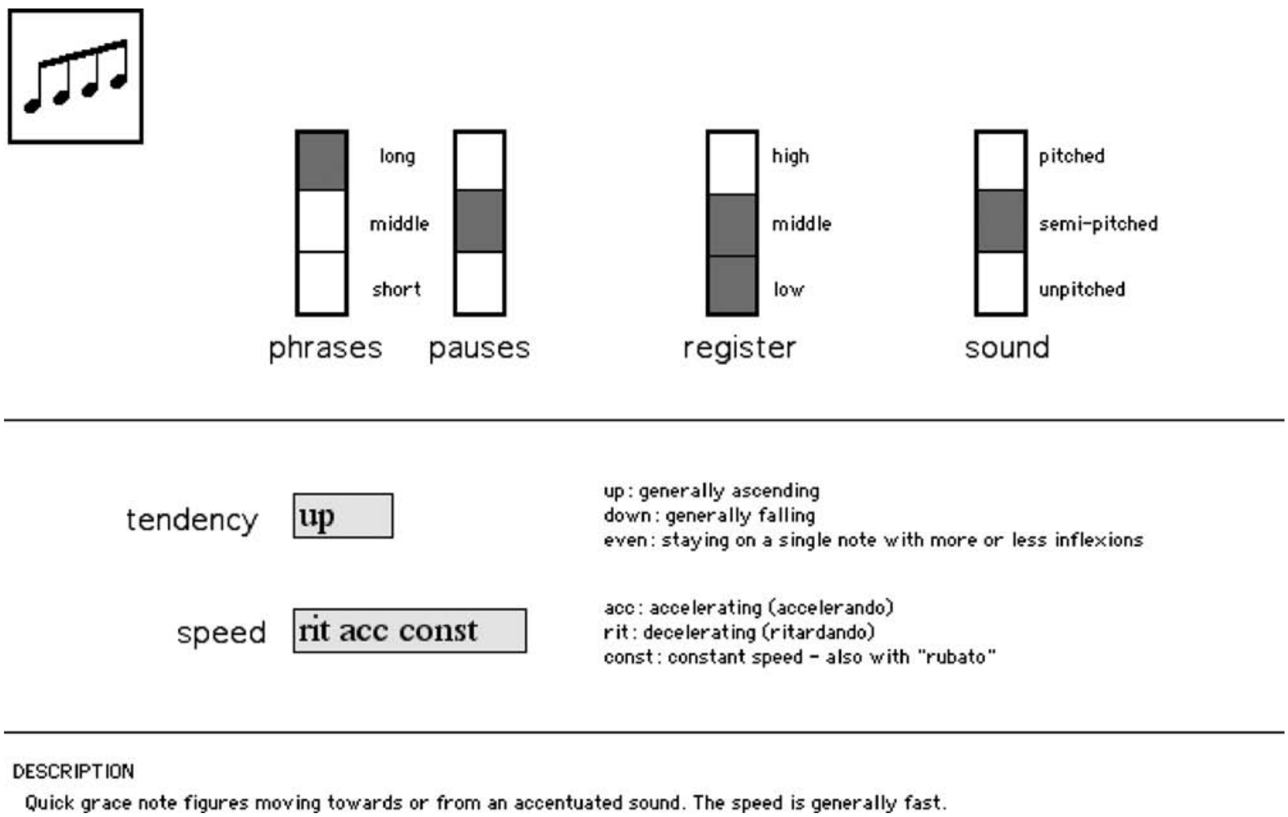
to indicate playing styles (such as clouds, points, trills, and drones); text specifying variations on those styles; and "global parameters" indicating phrase and rest durations, pitch registers, and timbre.

One or more conductors initiate triggers on a server. Each trigger toggles the state of a single player between tacet and active; tacet players preview a new notation segment, while active players begin improvising based on that segment. The conductors' role is simply to send triggers; stochastic software algorithms, operating autonomously, decide to whom a trigger is sent and how the trigger changes the notation.

Essl's work creates a feedback loop in which the algorithm's notation influences how the musicians play, the musicians' sound influences how the conductors trigger the algorithm, and the conductors' triggers influence the timing and pacing of the algorithm's notation. The audience follows this process by watching the conductors and listening to the musicians as they enter and exit the texture and change their improvisation styles. They do not directly perceive the software's output—only its effect on the music the ensemble plays.

The decision to use real-time notation in a composition does not preclude the use of electronic sound; many composers (e.g., Winkler 2004; Hajdu 2006) have used both in their works, essentially combining Figures 3a and 3b. The audience still hears a merged source—the musicians' interpretation of the real-time notation—but they hear it alongside electronic sound generated directly by the software. This combination creates new sonic possibilities, of course, but it can also detract from

Figure 4. Karlheinz Essl, Champ d'Action *(1998) for computer-controlled ensemble, excerpt from real-time music notation.*



the focus on a merged expression of algorithm and performer. Yet the use of real-time notation in this context enables composers to tightly integrate the acoustic and electroacoustic elements of their work in performance. Not only can the electronic sound respond to the musicians and the musicians to the electronic sound, but the score itself can respond to the dynamics of the performance.

Connecting Audiences and Performers

When audiences generate input into a real-time notation algorithm, and when the algorithm's output is interpreted by performing musicians, then a real-time notation system becomes a powerful link between the audience's participation and the musicians' performance. Such connections follow recent technological and aesthetic trends that have challenged us to become more engaged, active cultural consumers, helping to create the content we enjoy rather than serving as mere spectators. We curate the playlists we listen to, we collaboratively filter the media we watch, and we share the remixes we create. And new interfaces for musical expression have invited a broad public to explore their musicality, for example by plucking a networked string (Tanaka 2000), pushing jets of water in a fountain (Mann 2007), or moving through space (Rokeby 1998).

Most of these tools and works of art, however, focus on the creation of sound; conventional musical instruments and the musicians who play them remain absent. Some projects, such as Tod Machover's *Toy Symphony* (Jennings 2003), have combined professional musicians with non-musicians using specialized software and new musical interfaces. But these collaborations are the result of intense workshops, and the broader concert audience does not have the opportunity to spontaneously participate.

Figure 5. Jason Freeman, Glimmer *(2004) for chamber orchestra and audience participation. Audience members wave light sticks to influence the real-time music notation.*

Real-time notation systems, then, offer the opportunity to link the creative activities of listeners to conventional musical ensembles during live performance. This creates a feedback loop in which the audience influences the notation, the notation influences the performers, and the performers, in turn, influence the audience.

For example, in Kevin Baird's *No Clergy* (Baird 2005), computer software stochastically generates successive pages of conventional staff-based music notation for each performer in the chamber ensemble; the notation is displayed to each musician on individual laptop screens. Audience members use a Web interface to vote on parameter values that control the algorithm, such as mean values and variance for dynamics and durations. McAllister, Alcorn, and Strain (2004) developed a performance environment in which individual audience members create musical notation for each performer within the ensemble. Chosen audience members use a stylus to draw notation on a PDA's touch screen. The gestures are wirelessly transmitted to a server and rendered on a larger computer display for the corresponding musician to read.

Wulfson, Barrett, and Winter (2007) created *LiveScore*, in which gallery visitors adjust knobs on a MIDI controller with labels such as "sparseness," "pitchiness," and "stasis." A quartet of musicians plays proportional music notation that is stochastically generated based on those parameters; the notation is sent wirelessly from a server to a laptop in front of each musician. The audience is encouraged to walk around to view each musician's score and to take turns adjusting the MIDI controller.

My own recent work has also used real-time notation to connect audiences to instrumental performers through algorithms. In *Glimmer* (2004), for chamber orchestra (Freeman 2005, 2006), the audience of 200–600 members is divided into groups, each of which influences a corresponding section of the orchestra by waving four-inch battery-operated LED light sticks back and forth (see Figure 5). Four consumer-grade video cameras capture images of the entire audience and forward them to a computer for analysis. Software, written with Max and Jitter, pre-processes each frame, performing color-plane extraction, image masking, and threshold noise reduction. The software then determines the total amount of motion of light sticks in each group, using a feedback filter to create momentary motion trails in the image when sticks are waved, adding all pixels in the frame, and normalizing the sum.

Figure 6. Jason Freeman, Glimmer *(2004) for chamber orchestra and audience participation. Computer-controlled LED lights display the real-time notation to each musician.*

Each group's total amount of light-stick motion, along with the derivatives of these motion readings in relationship to other groups, determines when its linked musicians play and what those musicians play: their pitches, tempo, dynamics, and textures. Audience groups that coordinate their actions to create large-scale changes in their motion readings are rewarded; the group's musicians play more often, change their musical material more frequently, and become more prominent within the orchestra.

Real-time notation is communicated to each musician through a computer-controlled LED light tube mounted on each performer's music stand (see Figure 6). The lights are daisy-chained together to power and data supplies that relay the color commands they receive via Ethernet (UDP) from the computer software. The hue of the light indicates which note to play, and its brightness indicates the note's dynamic level. Short flashes of light indicate accents and cue musicians to prepare to play after a rest. In addition to the light, each musician has a single sheet of paper on the music stand. This is simply a reference to remind musicians of the fixed mappings from color to pitch, not printed notation to be read from start to finish.

Using light as music notation is both a practical communication protocol and an important visual component of the work. The dark space of the concert hall, punctuated by both audience and musician lights, helps create the sense of community necessary to facilitate the audience's participation. Furthermore, the ambient glow complements the sparse, slowly shifting harmonic landscapes of the music: As a player's music crescendos, their light brightens, and the player emerges from the darkness and into the audience's view.

In *Flock* (2007), a recent project of mine for saxophone quartet, up to 100 audience members, four dancers, and the four musicians walk around an open performance space, interacting with each other in accordance with simple instruction sets. A computer vision system utilizes a ceiling-mounted wide-angle Firewire camera to track the positions of each musician and audience member. The software,

Figure 7. Jason Freeman, Flock (2007) for saxophone quartet, excerpt from real-time music notation. Colors show which musician plays each gestural contour, horizontal dotted lines indicate pitches in the active set, and the thick vertical line scrolls to display measure position.

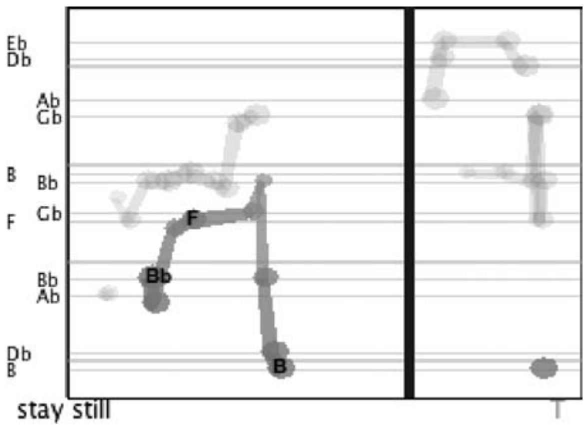


Figure 8. Jason Freeman, Flock (2007) for saxophone quartet. A PocketPC PDA is mounted onto a saxophone using standard marching-band lyres and flip-folder attachments to display real-time music notation.

written in Max and Jitter, first corrects the barrel distortion of the camera's fisheye camera lens and eliminates background noise with a threshold filter. Audience members, who wear lighted hats to facilitate accurate and efficient tracking, are identified with simple "blob" detection. The saxophonists, wearing hats lit in unique colors, are tracked using particle filters that use data from prior image frames to help stay with their color targets (Nummiaro, Koller-Meier, and Van Gool 2003).

Separate software components written in Java use the position data gathered by the camera and computer-vision software to generate real-time notation for each musician. The piece's real-time score incorporates graphical shapes and contours (see Figure 7), textual instructions, and staff-based notation in different sections of the piece. A flexible library of mappings, configurable via a graphical user interface by the composer during runtime, enables single people, groups, and their trajectories over time to be mapped to specific pitch sets and dynamic ranges. Analysis of individual and group velocity, acceleration, and physical spacing can also be mapped algorithmically to musical parameters such as dynamics, tempo, articulation, and clustering. The software uses both polar and Cartesian coordinate systems to map locations within the performance space, although these are always converted to a standard representation for notational display: Pitch remains on the vertical axis and time on the horizontal axis. A scrolling cursor indicates measure position and maintains sync among the players.

The computer software sends the real-time notation wirelessly to PocketPC PDA devices, which are mounted on each musician's saxophone using standard marching-band music lyres and flip-folder attachments (see Figure 8). Image and timing data for each measure are compressed into a byte stream, and the data is sent over an 802.11 wireless network to the PDA, where a Java application, written for CDC/Personal Profile with SWT, decompresses and parses the bytes, rendering the images and scrolling cursor on the display accordingly. The PDA software also maintains time sync with the desktop computer via a simple NTP-style protocol, using round-trip ping times to distinguish clock differences from network latency.

Flock uses two complementary strategies to minimize system latency while making best use of limited screen size, displaying only a single measure at a time. In one operation mode, best suited to dense textures, new data is sent once per measure. Measures begin rendering a second early, from left to right, so that performers can read ahead in the notation as they would with a printed score. Unfortunately, this necessitates the introduction of a significant latency into the system so that measures can be rendered well in advance. In an alternate mode, best suited to sparser textures, notation information is sent at regular intervals, several times per measure, and the entire measure is updated at once. System latency is much lower, but musicians can be caught by surprise when the score suddenly changes mid-measure.

Challenges in Designing Real-Time Notation for Performers and Audiences

Real-time notation systems offer exciting opportunities to rethink the relationships among algorithms, human performers, and audiences. They also present unique design challenges beyond those usually associated with interactive computer music performance. In this section, I explore two of the most interesting of these challenges: the design of a score that must be sight-read in performance, and the illustration to the audience of the role of the algorithm and notation in the performance.

Designing Scores for Sight-Reading

When music notation is generated on the fly during live performance, musicians have no opportunity to practice and rehearse the score in advance. Gerhard Winkler explains that "before and after the moment of performance, the piece—in the historical sense—does not exist" (Winkler 2004, p. 4). Musicians may practice other instances of the work but must ultimately sight-read in concert. Although this adds an exciting element of spontaneity, it also requires that musicians reconcile the dual challenges of accuracy and expressivity; they must not only play the score in front of them as it unfolds, but they must also "bring sense into this succession of unexpectable moments" to create a personal, coherent interpretation of that score (Winkler 2004, p. 4).

Through the design of their notation, composers must guide performers through these challenges, indicating which aspects of the score are to be played precisely and which are open to freer interpretation. As musicians practice a composition, their increasing familiarity with the elements of the notation should help them to perform it more accurately. But that familiarity should also lead them to develop a richer, more personal musical language with which to interpret it.

Some composers have addressed these challenges by limiting their real-time notation to a selection of pre-determined score excerpts. Performers do not know the order of the excerpts in advance, but they can practice each excerpt separately. In performance, they read the familiar fragments with relative ease and focus on stitching them together into a cohesive interpretation. David Kim-Boyle, for example, uses one-page score excerpts from eight famous piano works in *Valses and Etudes* for piano and computer (Kim-Boyle 2005). The software algorithm stochastically selects score pages and defines windows and trajectories across them to expose only a small, continuously moving segment of the page to the pianist at any moment.

Other composers have limited the complexity of real-time notation to make sure that, even if unfamiliar, it is still easy to sight-read in concert. In my work *Glimmer*, for example, I constrained the notation system to generate simple (though not simplistic) music in which each musician plays long, sustained notes on just a handful of different pitches during the piece. Although there were aesthetic motivations behind this design, it was also a practical recognition of the realities of contemporary orchestral performance: The ensemble had only one hour of rehearsal time to devote to the piece.

Many other works employing real-time notation, such as Karlheinz Essl's *Champ d'Action* (Essl 1998) and Art Clay's *GoingPublik* (Gutknecht, Clay, and Frey 2005), use unconventional graphical notation to guide the improvisation of performers by specifying pitch registers, rhythmic density, contours, and other more abstract information. In such works, the musicians, free from the need to accurately sight-read difficult passages in concert, can focus more on expressing themselves musically and creating cohesive large-scale phrases in consort with the rest of the ensemble. And when real-time systems use graphical notation in lieu of staff-based notation to represent dense musical passages, they also circumvent the limitations of automated notation layout algorithms, which must weigh flexible layout rules against graphic design sensibility (Dannenberg 1993).

My own work, *Flock,* uses a combination of fairly conventional staff-based notation and more abstract graphical notation. In some sections of the piece, the mappings from position data to notation are fairly straightforward, and it is important to me that the harmonic and rhythmic structures, along with those mappings, come through clearly. Because the music in these passages is relatively sparse, conventional staff-based notation is not challenging to sight read, and its precision limits improvisation, keeping the harmonic and rhythmic structures well intact. In other sections, the density and complexity of musical material poses practical challenges both for automated notation layout and for sight-reading, and the mappings, which incorporate the movement trajectories of dozens of people over several seconds, are inherently less transparent; it is consequently not important to me that every pitch and rhythm be performed precisely. In these passages, graphical notation (see Figure 7) groups notes together into longer melodic contours and gestures, indicating a handful of structurally important pitches but leaving the exact content to the musicians' improvisation.

As musicians prepare to perform these kinds of pieces, Gerhard Winkler notes that the process "shifts from 'studying notes' to . . . [getting to know] . . .'how the system works'" (Winkler 2004, p. 4). Percussionist Tom Sherwood recalls his experience rehearsing Jennifer Walshe's *meanwhile, back at the ranch . . .* (Walshe 2005), which employs real-time graphical notation. The ensemble initially concentrated on learning the notation's symbolic vocabulary and then focused on creating expressive music as a group: "It would have been less interesting if we were totally at the mercy of the notation. But once we got familiar with the process and developed a common approach to the notation . . . then it became more musical" (Sherwood 2008). Saxophonist Jason Kush, who has performed *Flock*, had a similar experience. Once he became comfortable reading the notation, he began to develop a more diverse vocabulary of musical gestures to respond to it, taking it "less literally . . . [and] . . . painting more of a sonic picture of what was up there." The resulting music, he felt, became more cohesive as he learned to better read ahead in each measure to form phrases, and as he learned the vocabulary of the other musicians and could start to predict what they would play (Kush 2008).

In contrast to these works, Nick Didkovsky's *Zero Waste*, for piano and computer (Didkovsky 2002, 2004), uses the challenge of sight-reading real-time notation as the foundation of the composition's design. *Zero Waste* requires the performer to sight-read complex, even unplayable, music as the audience follows along with the score on a projection screen. Not only is failure expected, it is demanded; the pianist's mistakes, in conjunction with the transcription algorithm's limitations and the quantization compromises of the conventional notation (Nauert 1994), give rise to the work's unique formal structure.

Mr. Didkovsky's software, written in his Java Music Specification Language (Didkovsky and Burk 2001), stochastically generates two initial measures of music. The software transcribes subsequent two-measure blocks of music from the MIDI data generated by the pianist's sight-reading performance; these become the next two measures of score notation to sight-read. The performer must use a piano that has MIDI output (whether built in, as in the Yamaha Disklavier, or supplied by an external sensor, such as the Moog PianoBar). The composer explains:

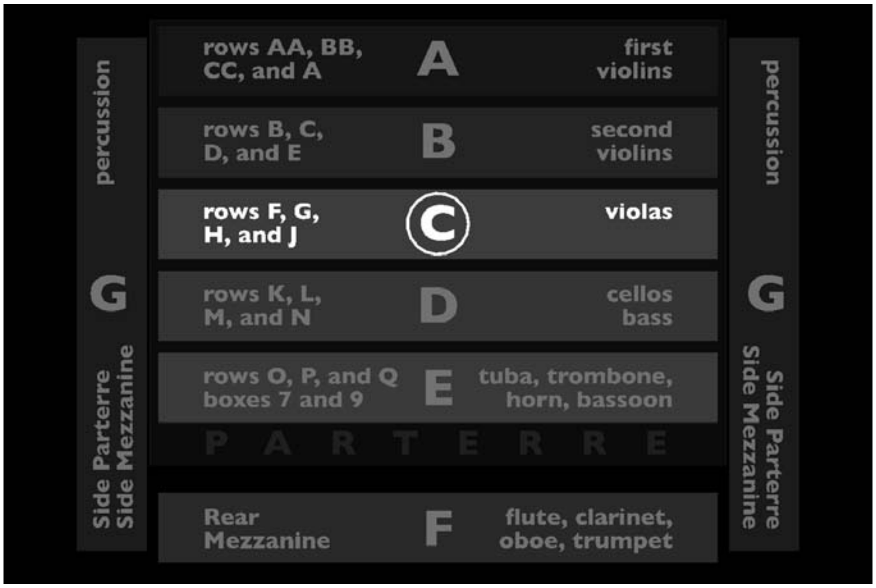
> If the performer were perfect, and if music transcription and notation were both theoretically and practically perfect, then *Zero Waste* would consist of identical repetitions of the first two measures. Of course, no sight reader is perfect, and notation must strike a balance between readability and absolute accuracy, so each new pair of measures diverges and evolves a bit more from the last (Didkovsky 2004, p. 746).

Notes and rests disappear, emerge, combine into chords and split into melodies as the initial motive gradually turns into new musical material.

Illustrating to the Audience

The Metropolitan Opera House in New York sells score desk tickets to students, who may use a desk

Figure 9. Video still from Jason Freeman, Glimmer *(2004) for chamber orchestra and audience participation.*



in front of their seat—complete with reading light—to follow along with a printed musical score of the opera during its performance. But beyond such unique circumstances, audiences rarely view the notation of a composition during its performance; they instead focus on the immediate aural and visual experience of the concert.

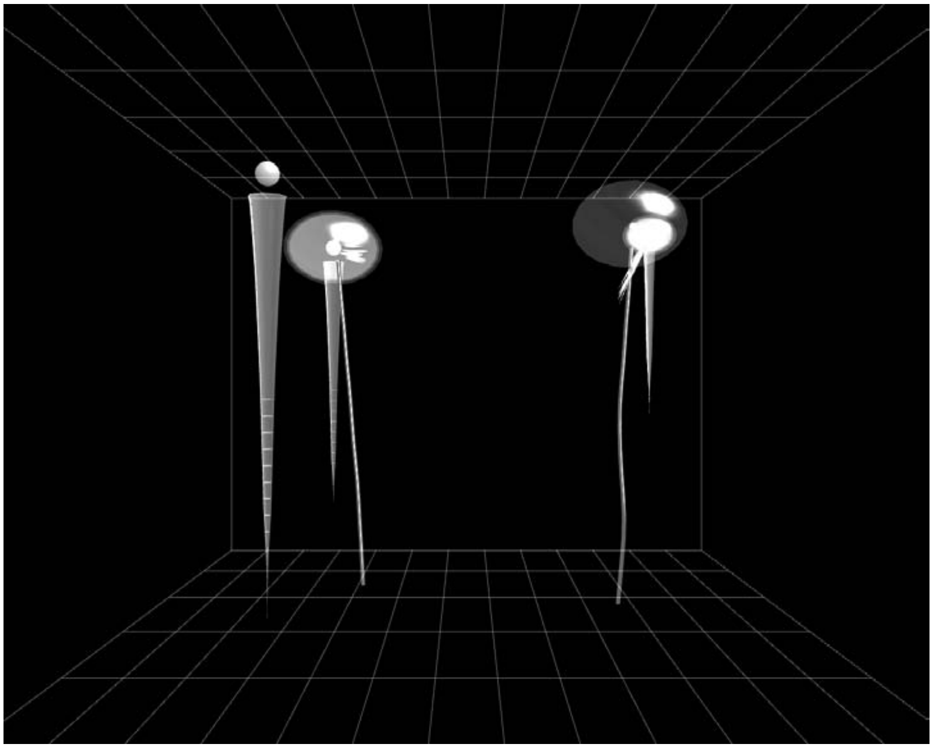
When algorithms and musicians interact with each other during performance, however, audiences often want to understand the role of those algorithms and the process through which they shape the music. In live performances involving new hardware and software-based instruments, many practitioners attempt to create a "clarity of interaction" through which audiences can understand not only the mapping of input gestures to sound output but also the ways in which these gestures function within the "sonic and structural space of the music" (Tanaka 2000, p. 399).

With real-time notation systems, the algorithm and human performer together create a single, merged sonic output. Although such merged expression is one of the compelling opportunities offered by real-time notation, it can also make it difficult for audiences to understand the role of the algorithm and its notation in the performance, because they perceive it only indirectly.

To address this challenge, many composers display the notation generated by their systems not only to the musicians but also to the audience, so that the audience can better understand the relationship between the algorithm's direct output and the musicians' interpretation of that output. Nick Didkovsky projects the notation for *Zero Waste* onto a large screen for the audience (Didkovsky 2004); Gerhard Winkler does the same in *KOMA* (Winkler 2004). But Mr. Didkovsky's score requires that audiences read musical notation, and several elements of Winkler's score draw from conventional Western notation as well. In ensemble works, it may be impractical, or at least visually overwhelming, to display every musician's part to the audience.

In my own works employing real-time notation, audiences help create the notation through their participation, and so it is particularly critical that they understand the output of the system, regardless of whether they can read music. In *Glimmer*, the audience sees the real-time notation directly. However, it is difficult to absorb data displayed by an orchestra of independently changing lights, so a simple video animation (see Figure 9) visualizes basic aspects of the algorithm's state. The display's layout mimics the seating chart of the concert hall, identifying each group of audience members and the

Figure 10. *Video still from Jason Freeman,* Flock *(2007) for saxophone quartet. The animation, created by Liubo Borissov, provides an alternate visualization of the notation data for the audience.*



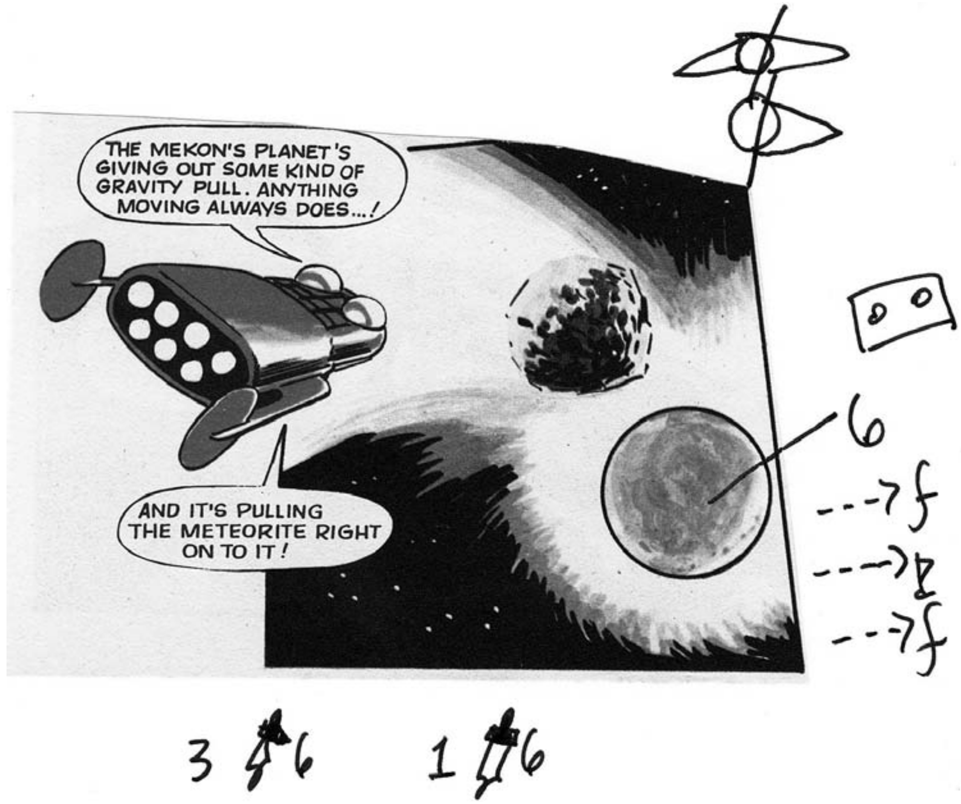
corresponding musicians they influence. The more an audience group waves its light sticks, the brighter their section of the display becomes. The animation also monitors the competitive aspect of the piece, highlighting the first-place audience group and hiding losing groups as their musicians are muted. In performances of the work, the visualization has helped audiences better understand their role in shaping the music they hear, but with one negative consequence: Some audience members have become so obsessed with the competitive elements emphasized by the animation that the music itself has been relegated to background listening for them.

In *Flock*, audience members cannot see the real-time notation unless they look over the shoulder of a musician, and, owing to latency and polar-to-Cartesian transforms, that notation is not always easy to understand at a glance. Instead, audience members see a multi-screen video animation by visual artist Liubo Borissov (see Figure 10) that renders position and notation data. Each video screen presents a three-dimensional view of the performance space with colored dots for each saxophonist and white dots for each audience member and dancer. Measure time moves up along the vertical axis, and in several sections of the piece each participant's movement over time also leaves a trail along this axis. When a saxophonist plays notes or gestures, a bubble momentarily surrounds that performer's dot, and a diagonal line connects the saxophonist to the participant who generated that note.

In performances of the work, the video has been crucial in showing audience members the role they play. As they enter the performance space, many audience members point at their dots on screen and move back and forth as they watch their dots follow them. And in sections where the musical mappings are less transparent, the animation continues to show position data as its base layer, and audiences continue to watch for these visual cues.

But in post-performance discussions with audiences, I have discovered that the video animation is not enough, on its own, to satisfy their curiosity.

Figure 11. Jennifer Walshe, *meanwhile back at the ranch . . .* (2005), for ensemble and image controller, excerpt from real-time music notation.



Many people have wanted to see the musicians' notation directly in order to understand first-hand the challenges the musicians face and the role the musicians play in interpreting the score. In upcoming performances of *Flock*, I am considering incorporating views of the real-time notation into the video projection, alongside Liubo Borissov's animation, so that audiences can benefit from the simplicity of the stylized animation but also feel more connected to the musicians' performance.

Jennifer Walshe's recent work *meanwhile, back at the ranch . . .* (Walshe 2005) offers an elegant approach to illustrating real-time notation to the audience. Rather than separating the design of notation for performers and of visualization for the audience, Walshe's notation is designed to be accessible to both. In the piece, one of the performers acts as an "image controller" to create real-time notation for seven chamber musicians. That performer draws, manipulates, and annotates images—often extracted from comic books and graphic novels—that are projected onto a large screen. The printed musical score reads like a syntax manual for a programming language, explaining how the image controller can annotate the images to guide the musicians' improvisation, using symbols to indicate everything from instrumentation, dynamics, and pitch registers to repetitions, transitions, and even superhero powers, which enable individual players to make autonomous decisions. The image controller selects the images and annotates them live during each performance (see Figure 11), directly creating the notation without any mediation by software or algorithms. Walshe's work appeals to audiences because of the immediacy and humor the images convey, because of the connection audiences feel when the see the same notation as the musicians and follow their struggle to perform it as music, and because of the adventure of figuring out, little by little, what all of the cryptic annotations to the images signify. Sound and images combine into a cohesive multimedia performance, because both express the same "deep structure" (Dannenberg 2005).

Conclusion

If one of notation's main purposes is to facilitate communication (Bent et al. 2007), then real-time notation enables new ways for a variety of constituents—from musicians, composers, and audiences to algorithms and chance procedures—to communicate with each other to shape a live musical performance. Notation becomes a vehicle for expressing the uniqueness of each performance of a work rather than a document for capturing the commonalities of every performance of that work.

In the realm of interactive computer-music performance, real-time notation offers an important tool within a larger arsenal of techniques. Todd Winkler notes that in interactive music, "the computer's capabilities . . . [can be] . . . used to create new musical relationships that may exist only between humans and computers in a digital world" (Winkler 1998, pp. 2–3). Joel Chadabe writes: "The challenge for computer music composers . . . will be to use their elite knowledge and skill to create situations in which members of the public without that knowledge and skill can participate meaningfully in a musical process" (Chadabe 2000, p. 11). Within these contexts, notation serves as an important link between the digital domain and the physical world and among composer, performer, and listener.

As the practice of real-time notation continues to grow, and as enabling technologies continue to improve, the types of communication that real-time notation systems facilitate will continue to expand. Because of the low bandwidth and distributed architecture of most real-time notation systems, they are particularly suitable for adaptation to multi-location performances, massively distributed online environments, and more general frameworks for remote musical collaboration. Performance systems such as Georg Hajdu's Quintet.net (Hajdu 2006), along with commercial products such as the eStand digital music stand, which can share annotations over peer-to-peer networks, suggest possibilities for real-time notation in networked musical contexts.

Although music notation serves primarily as a pragmatic tool for communication, musical scores also exist as visual entities. Manuscripts by Monteverdi sit within the vaults of rare-book archives, sketches by Beethoven are sold at art-house auctions, and scores by John Cage hang on museum walls. Concrete scores by composers ranging from Baude Cordier to Charles Ives to George Crumb visually depict programmatic elements in the music through novel graphic design. Real-time notation systems combine this tradition with more recent trends in multimedia and interactive performance, using new technologies to move score generation from the roof of La Scala down to its main stage.

Acknowledgments

Glimmer was commissioned by the American Composers Orchestra, with additional support from the American Music Center's Composer Assistance Program and Meet the Composer's Creative Connections Program. *Flock* was commissioned by the Adrienne Arsht Center for the Performing Arts of Miami-Dade County, with additional support from the Funding Arts Network, the Georgia Tech Foundation, and the GVU Center at Georgia Tech. Fernando Benadon and Mark Godfrey provided invaluable feedback on early drafts of this article.

References

Ames, C. 1987. "Automated Composition in Retrospect: 1956–1986." *Leonardo* 20(2):169–185.

Assayag, G. 1998. "Computer Assisted Composition Today." *First Symposium on Music and Computers*. Corfu, Greece: Ionian University. Available online at recherche.ircam.fr/equipes/repmus/RMPapers/Corfou98.

Assayag, G., et al. 1999. "Computer-Assisted Composition at IRCAM: From PatchWork to OpenMusic." *Computer Music Journal* 23(3):59–72.

Baird, K. 2005. "Real-Time Generation of Music Notation via Audience Interaction Using Python and GNU Lilypond." *Proceedings of the 2005 International*

Conference on New Interfaces for Musical Expression. New York: Association for Computing Machinery, pp. 240–241.
Bent, I., et al. 2007. "Notation, §I, 1: General: Introduction." *Grove Music Online*. Available online at www.grovemusic.com/shared/views/article.html?section=music.20114.1.1. Accessed 13 June 2007.
Brown, E. 1961. *Available Forms I*. New York: Schirmer.
Brown, E. 1966. *Calder Piece*. Rye, New York: The Earle Brown Music Foundation.
Brown, E. 1999. "Transformations and Developments of a Radical Aesthetic." *Current Musicology* 67/68: 39–57.
Chadabe, J. 2000. "Remarks on Computer Music Culture." *Computer Music Journal* 24(4):9–11.
Cope, D. 1996. *Experiments in Musical Intelligence*. Middleton, Wisconsin: A-R Editions.
Dannenberg, R. 1993. "A Brief Survey of Music Representation Issues, Techniques, and Systems." *Computer Music Journal* 17(3):20–30.
Dannenberg, R. 2005. "Interactive Visual Music: A Personal Perspective." *Computer Music Journal* 29(4): 25–35.
Didkovsky, N. 2002. *Zero Waste* (score and recording). Available online at www.punosmusic.com/pages/zerowaste/index.html. Accessed 17 June 2007.
Didkovsky, N. 2004. "Recent Compositions and Performance Instruments Realized in the Java Music Specification Language." *Proceedings of the 2004 International Computer Music Conference*. San Francisco, California: International Computer Music Association, pp. 746–749.
Didkovsky, N., and P. Burk. 2001. "Java Music Specification Language: An Introduction and Overview." *Proceedings of the 2001 International Computer Music Conference*. San Francisco, California: International Computer Music Association, pp. 123–126.
Essl, K. 1998. *Champ d'Action*. Available online at www.essl.at/works/champ.html. Accessed 17 June 2007.
Freeman, J. 2005. "Large Audience Participation, Technology, and Orchestral Performance." *Proceedings of the 2005 International Computer Music Conference*. San Francisco, California: International Computer Music Association, pp. 757–760.
Freeman, J. 2006. "Glimmer: Creating New Connections." *Proceedings of Digital Art Weeks*. Zürich: Die Eidgenössische Technische Hochschule (ETH). Available online at www.digitalartweeks.ethz.ch/docs/daw06proc/papers/freeman-glimmer-color.pdf.
Garnett, G. 2001. "The Aesthetics of Interactive Computer Music." *Computer Music Journal* 25(1):21–33.
Gutknecht, J., A. Clay, and T. Frey. 2005. "GoingPublik: Using Realtime Global Score Synthesis." *Proceedings of 2005 International Conference on New Interfaces for Musical Expression*. New York: Association for Computing Machinery, pp. 148–151.
Hajdu, G. 2006. "Automatic Composition and Notation in Network Music Environments." *Proceedings of Sound and Music Computing 2006*. Marseilles, France: Centre National de Création Musicale. Available online at www.quintet-net.org/info/Automatic_Composition.pdf.
Hiller, L. 1981. "Composing with Computers: A Progress Report." *Computer Music Journal* 5(4):7–21.
Hiller, L., and L. Isaacson. 1957. *Illiac Suite for String Quartet*. New York: New Music Edition.
Hiller, L., and L. Isaacson. 1959. *Experimental Music: Composition with an Electronic Computer*. New York: McGraw-Hill.
Hughes, S. 1956. "How to Write an Overture: A Rossini Recipe." *Musical Times* 97(1359):247–249.
Jennings, K. 2003. "'Toy Symphony': An International Music Technology Project for Children." *Music Education International* 2:3–21.
Kim-Boyle, D. 2005. "Musical Score Generation in *Valses and Etudes*." *Proceedings of the 2005 International Conference on New Interfaces for Musical Expression*. New York: Association for Computing Machinery, pp. 238–239.
Kim-Boyle, D. 2006. "Real-Time Generation of Open-Form Scores." *Proceedings of Digital Art Weeks*. Zürich: Die Eidgenössische Technische Hochschule (ETH). Available online at www.digitalartweeks.ethz.ch/docs/daw06proc/posters/kim_boyle-open_form_scores.pdf.
Kush, J. 2008. Personal interview, 15 February.
Kuuskankare, M., and M. Laurson. 2006. "Expressive Notation Package." *Computer Music Journal* 30(4):67–79.
Laske, O. 1981. "Composition Theory in Koenig's Project 1 and Project 2." *Computer Music Journal* 5(4): 54–65.
Mann, S. 2007. "Natural Interfaces for Musical Expression: Physiphones as Primordial Infra-Instruments." *Proceedings of the 2007 International Conference on New Interfaces in Musical Expression*. New York: Association for Computing Machinery, pp. 118–123.
McAllister, G., M. Alcorn, and P. Strain. 2004. "Interactive Performance with Wireless PDAs." *Proceedings of the 2004 International Computer Music Conference*. San Francisco, California: International Computer Music Association, pp. 702–705.

Morales-Manzanares, R. 1992. "Non-Deterministic Automatons Controlled by Rules for Composition." *Proceedings of the 1992 International Computer Music Conference*. San Francisco, California: International Computer Music Association, pp. 400–401.

Nauert, P. 1994. "A Theory of Complexity to Constrain the Approximation of Arbitrary Sequences of Timepoints." *Perspectives of New Music* 32(2): 226–263.

Nummiaro, K., E. Koller-Meier, and L. Van Gool. 2003. "An Adaptive Color-Based Particle Filter." *Image and Vision Computing* 21(1):99–110.

Roads, C. 1996. *The Computer Music Tutorial*. Cambridge, Massachusetts: MIT Press.

Rokeby, D. 1998. "The Construction of Experience: Interface as Content." In C. Dodsworth, ed. *Digital Illusion: Entertaining the Future with High Technology*. Boston, Massachusetts: ACM Press/Addison-Wesley, pp. 27–48.

Rothstein, E. 1981. "Calder Mobile Conducts a Work by Earle Brown." *The New York Times*, C18, col. 3, 16 November.

Sherwood, T. 2008. Personal interview, 13 February.

Stockhausen, K. 1957. *Klavierstück XI*. London: Universal Edition.

Tanaka, A. 2000. "Musical Performance Practice on Sensor-based Instruments." In M. M. Wanderley and M. Battier, eds. *Trends in Gestural Control of Music*. Paris: IRCAM, pp. 389–406.

Walshe, J. 2005. *meanwhile, back at the ranch*. Dublin: Milker Corporation.

Winkler, G. 2004. "The Realtime-Score: A Missing-Link in Computer-Music Performance." *Proceedings of Sound and Music Computing 2004*, Paris: IRCAM. Available online at recherche.ircam.fr/equipes/repmus/SMC04/scm04actes/P3.pdf.

Winkler, T. 1998. *Composing Interactive Music: Techniques and Ideas Using Max*. Cambridge, Massachusetts: MIT Press.

Wulfson, H., G. Barrett, and M. Winter. 2007. "Automatic Notation Generators." *Proceedings of the 2007 International Conference on New Interfaces in Musical Expression*. New York: Association for Computing Machinery, pp. 346–351.

Xenakis, I. 1992. *Formalized Music*. Stuyvesant, New York: Pendragon Press.