Real-Time Music Notation, Collaborative Improvisation, and Laptop Ensembles

Sang Won Lee
Center for Music Technology
Georgia Tech.
840 McMillan Street
Atlanta, GA 30332 USA
sangwonlee717@gmail.com

Jason Freeman
Center for Music Technology
Georgia Institute of Technology
840 McMillan Street
Atlanta, GA 30332 USA
jason.freeman@gatech.edu

Andrew Collela
Center for Music Technology
Georgia Institute of Technology
840 McMillan Street
Atlanta, GA 30332 USA
ajcolella@gmail.com

ABSTRACT

This paper describes recent extensions to LOLC, a text-based environment for collaborative improvisation for laptop ensembles, which integrate acoustic instrumental musicians into the environment. Laptop musicians author short commands to create, transform, and share pre-composed musical fragments, and the resulting notation is digitally displayed, in real time, to instrumental musicians to sight-read in performance. The paper describes the background and motivations of the project, outlines the design of the original LOLC environment and describes its new real-time notation components in detail, and explains the use of these new components in a musical composition, SGLC, by one of the authors.

Keywords

Real-time Music Notation, Live Coding, Laptop Orchestra

1. INTRODUCTION

This paper describes recent extensions to LOLC[11], a text-based environment for collaborative improvisation for laptop ensembles, which integrate acoustic instrumental musicians into the LOLC environment. LOLC's text commands and networked development environment originally enabled the creation, transformation, and sharing of short musical motives built from short pre-recorded audio files. The extensions described in this paper adapt the same environment to instead display, transform, and share short fragments of pre-composed music notation. The resulting real-time notation is digitally displayed to instrumental musicians, who sight-read it live in concert as it is created by the laptop performers.

Our goals in integrating real-time music notation into LOLC were two-fold. First, we sought to explore a new model for integrating acoustic musicians into laptop ensembles, deeply involving the instrumentalists in the collaborative, conversational style of improvisation that LOLC facilitates. Second, we wished to provide an accessible environment within which musicians could explore real-time notation and its possibilities without the need for advanced technical skills, since few such tools are yet available in this rapidly developing field[9].

This paper describes the background and context in which LOLC and its real-time notation components were developed; briefly overviews the original LOLC environment; describes the design and implementation of the real-time notation extensions; and discusses a new performance piece, SGLC, created with these extensions.

2. BACKGROUND

LOLC and its real-time notation extensions were inspired by prior research and creative work in multiple areas. Computer music languages such as Impromptu[2] and extensions to existing languages, such as JITLib for SuperCollider[5] and the Co-Audicle for ChucK[18] facilitate collaborative live coding in networked laptop ensembles through features such as clock synchronization, shared objects, shared code, and chat. Earlier networked music groups also provided models that directly influenced LOLC: in the Hub's[3] work *Borrowing and Stealing*, for example, players stored symbolic representations of musical motives in a shared data store and manipulated those contributed by other musicians.

LOLC was also influenced by collaborative improvisation in other types of ensembles. Composers such as John Zorn[19] and George Lewis[15] have developed structured strategies for group improvisation. And ethnomusicologists have characterized group interaction among improvising jazz musicians as "borrowing material from one another and transforming it"[1].

The real-time notation components of LOLC are based on recent research and creative work in the field of real-time music notation[8]. This field in turn draws from the long histories of algorithmic and computer-assisted symbolic and notated compositions, with the *Illiac Suite* as an early example[12], and from open-form scores by composers such as Brown[4] and Stockhausen[19], in which performers read the printed score differently in each unique performance. With the proliferation of fast, small, networked computers and digital displays, the open score can become digital and increasingly malleable, and real-time algorithms can generate and render that score live in concert. The entire process can then potentially react to external inputs and interfaces, whether from the performer(s), the composer, the audience, or another external actor. Recent examples of real-time notation works include Essl's *Champ d'Action*[7], Rebelo's *Playspace*[16], and Freeman's *Flock* [10].

3. LOLC

In LOLC[11], the musicians in a laptop ensemble use a networked, textual performance interface to create and share rhythmic motives based on a collection of recorded sounds. (LOLC was presented as a performance at NIME 2011.) The environment encourages musicians to share their code with each other, developing an improvisational conversation over time as material is borrowed and transformed. LOLC was designed to facilitate collaborative improvisation by large laptop ensembles whose members may be technical novices.

The environment is thus deliberately simple and constrained, and we do not consider LOLC to be a programming language.

The LOLC software environment comprises server software that synchronizes pattern definitions across the ensemble and maintains time synchronization; client software that presents an integrated development environment (IDE) interface for text-based performance and collaboration; and a visualization of performers' activities projected for the audience to watch.

In the original version of LOLC, pre-recorded one-shot sound files served as the musical building blocks. Performers defined rhythmic patterns based on repetition of the sounds with specified durations and dynamics:

```
mySound: "a1.wav"
myPattern: mySound[e.ff, q.ff, s.n, s]
```

This example creates an eighth and quarter note at fortissimo, a sixteenth-note rest, and one sixteenth note at the default mezzo-forte.

Patterns can also be compressed and stretched to created nested patterns:

```
myNested: myPattern[w,h]
```

This example plays myPattern twice. The first time, each note value is doubled so that the pattern, which was originally two beats long, fills the space of a whole note; the second time, it is played back in its original form.

Patterns can also be transformed through thirteen unary and binary operations such as reversing, shuffling, concatenating, shuffling, and truncating. The syntax is straightforward:

```
myConcat: cat(pattern1, pattern2)
myTrunc: trunc(pattern2, 2)
```

Patterns are not played immediately upon creation. Instead, musicians explicitly schedule them for playback at a particular point in time for a particular number of repetitions:

```
play myPattern @nextMeasure
loop myPattern @236 ~4
```

The first example plays the pattern a single time at the beginning of the subsequent measure; the second plays the pattern four times beginning at measure 236.

Collaboration is built into LOLC. Commands are typed into an instant-messaging-style interface that shows both commands and chat messages from everyone in the ensemble (Figure 1). As musical patterns are created, they are automatically shared with the other musicians and displayed in a pattern library. Players cannot modify existing patterns in the library, but instead create derivative patterns with new names. This ensures that pattern names map consistently to the same musical content throughout the performance.

4. REAL-TIME NOTATION IN LOLC

In extending LOLC to the realm of real-time music notation, we strove to keep the environment, including the syntax and command structure, as consistent as possible. The substantial changes, rather, are to the system's input and output. Instead of building musical patterns from audio files, commands operate on pre-composed musical score files. Instead of outputting electroacoustic sound over speakers, the system renders notation to an external computer monitor. Each laptop musician generates music notation for a single paired instrumental musician. The two musicians sit side by side in performance. (Figure 3)

This approach addressed one of the key challenges of real-time music notation systems — facilitating sight-reading in concert — by adopting the common solution of building upon pre-composed musical fragments[13]. While these fragments can eventually be transformed in performance into completely unrecognizable material, they provide an important foundation of known notation that instrumental musicians can practice. Sight-reading is further facilitated by limiting notation to monophonic scores, though this does limit the instruments best suited for use with the environment.

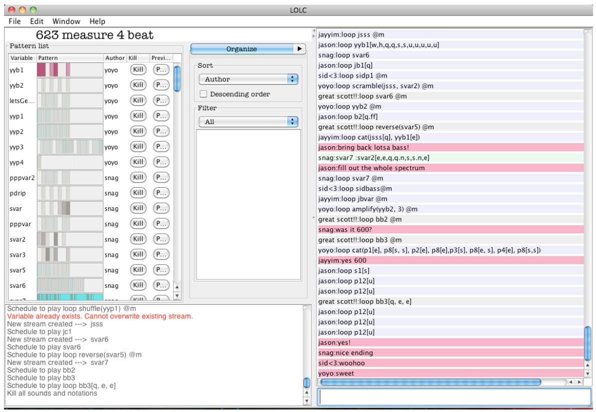

Figure 1. The LOLC client application.

The integration of real-time notation and instrumental musicians into LOLC creates new types of collaborations within the ensemble: the instrumental players respond to the real-time notation, to each other, and to the text chat messages displayed on their screens; the laptop musicians respond to the instrumentalists' musical interpretations of (and sometimes improvisations guided by) the notation they have created and to each other through chatting, visualization, and sound; and the audience glimpses the thought processes of the laptop musicians, the instrumentalists' interpretations, and the composite result. It also emphasizes the uniqueness, excitement, and risk of live improvisation in a new way: not only do the laptop musicians type text live in performance to create the music, but the instrumentalists also sight-read the results.

4.1 Real-time Notation Design

We wanted to keep the LOLC syntax for notation as close as possible to that for digital audio so that laptop musicians who learned LOLC in one context could immediately approach the other and could potentially combine both notation and audio in a single performance. This section discusses the discrepancies between the two domains and the reasons behind those changes.

Instead of loading digital audio files, LOLC loads symbolic notation files. Notation files are not expected to be one-shot as with the audio files, but may have any duration, from a single note to many measures, of a monophonic musical motive. We believe that most laptop performers would find it too time-consuming to create scores note by note in performance and that most instrumentalists would find it too difficult to sight-read the resulting notation. (That approach, though, is supported through the use of single-note score files.) Permitting longer score files gives both constituencies an important starting base from which to perform. Through the operations supported in LOLC, laptop musicians can also quickly transform, recombine, and extend single notes or pre-composed motives using simple pre-defined algorithms.

The bracket syntax to create and nest patterns remains the same as with audio, but the results are slightly different. Instead of stretching or compressing a score file or pattern to match the specified duration, LOLC simply truncates the score to match the desired duration. We were concerned that otherwise the rhythmic density of scores would quickly become too complex for sight-reading.

We support all of the same transformation operations for both audio and notation. For notation, though, we added an additional parameter to specify whether operations proceed at the hierarchical level of notes (default), measures, or hypermeasures (a group of four measures). For example, a reverse operation could reorder the notes from last to first or

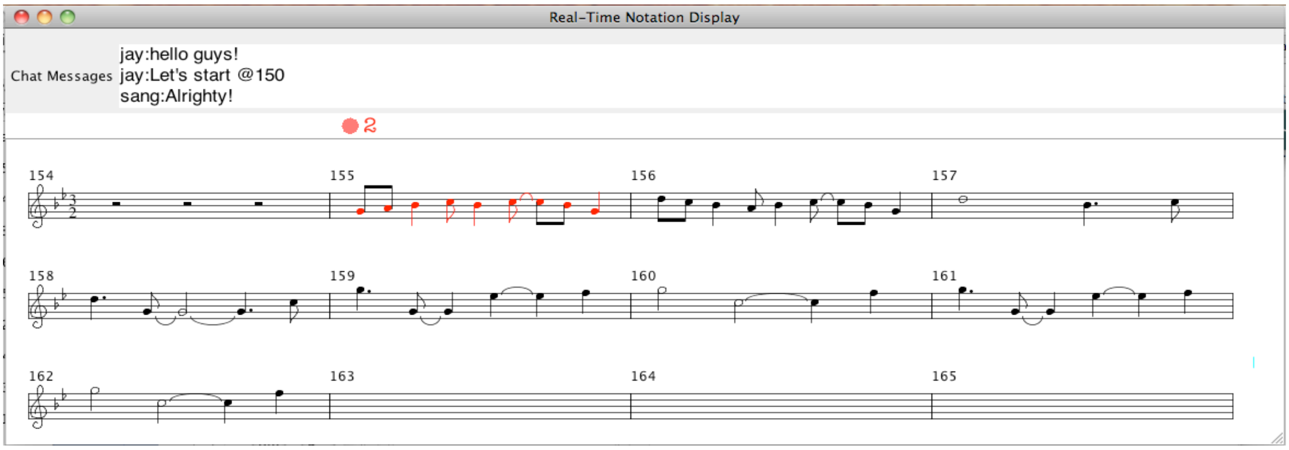

Figure 2 Notation Viewer and Beat Indicator

it could reorder the measures from last to first but keep the notes within each measure in their original order. We found this feature important because in practical use, the duration of audio patterns in LOLC rarely exceeds a measure while the duration of notation patterns almost always does. This parameter therefore gives laptop musicians an important level of control over how operations are applied that was not previously necessary. We also added a new operation, transpose, that we had not found important with audio, where the content tended to be more percussive and less pitched.

Scheduling notation patterns for display follows the same syntax as scheduling audio patterns for playback, but again, the behavior is slightly different in each case. Audio scheduling supports simultaneous playback of multiple patterns, but notation does not for obvious playability reasons: newly scheduled patterns overwrite any previously scheduled content in the measures for which they are scheduled. Also, LOLC enforces scheduling notation in advance, so that instrumentalists will have time to prepare to play the rendered music. With audio, patterns can be scheduled as little as a single beat in the future, but with notation, patterns are scheduled at earliest a full measure in advance and always begin at the start of a measure.

4.2 Technical Implementation

The LOLC client and server software are written in Java. To add real-time music notation, we utilized the Java Music Specification Language (JMSL) API[6], which supports real-time rendering of Western staff-based notation with limited graphical extensions. We use JMSL's XML file format for scores as the native notation format for LOLC. Composers can use JMSL's GUI score editor application to create these or to import them from MIDI files, or they can be created algorithmically.

Notation is displayed by each LOLC client application on an external monitor directly connected to the laptop. While we explored the use of wireless displays and networked tablets, this solution proved the best for LOLC, since there is already a one-to-one mapping of laptops to notation displays and since each instrumental musician should be seated next to the laptop musician who creates her notation. With this setup, the laptop musician interacts with the main LOLC interface window on her built-in display and the entire external monitor is dedicated to notation display (Figure 2). The monitor is positioned so that the instrumental musician can read it comfortably and the laptop musician can glance over at it occasionally.

The notation display includes several staves of notation, showing about 20 measures of lookahead. As the music reaches the end of the first staff system, the entire display scrolls so that the next staff system moves to the top. In this manner, musicians always read the uppermost staff system and the remaining systems solely show the music coming next. The current beat and measure is indicated through a moving display that always hovers above the current measure. It includes both a numerical beat indicator and a circle that pulses and bounces to indicate each beat. (Performances with LOLC are intended to be unconducted, so this display provides critical synchronization information for the instrumental musicians.) Additionally, all the notes in the current measure are highlighted in red, and if the measure is otherwise empty, it is filled with red rests.

A configuration dialog box lets users set various display parameters: display clef, transposition (either chromatic or key signature) for transposing instruments, instrument range (outside of which notes will be octave-transposed to fit within the range), and zoom level for score display.

5. SGLC

The original digital-audio version of LOLC existed in a gray space between a musical composition and a software framework. The features and limitations of the environment pushed improvising musicians towards particular modes of performance and collaboration, but the open-ended structure of the performance and the openness of the system to new audio content belied any identity as a single musical composition. Our performance guide for the digital audio version of LOLC leaves decisions about audio content, performance structure, and musical roles up to the ensemble.

With the introduction of real-time notation, LOLC's position along this continuum moves more towards composition, or rather towards multiple compositions that are each defined, at a minimum, by a set of pre-composed fragments of musical notation.

The first such composition for LOLC and real-time notation is SGLC, composed by one of the authors (Freeman). Its compositional identity is defined by three components: a collection of pre-composed fragments; a musical structure indicating at a broad level how the laptop musicians are to work with those fragments over the course of the performance; and timing specifications about meter (3/2), tempo (84 bpm) and overall duration (approximately 250 measures). The instrumentation remains open.

The 53 pre-composed musical fragments are each between one and three measures in length and fall into four categories: long-tones, contrapuntal melodies, graphical contours, and textual instructions. The former two categories are strictly notated, while the latter two guide improvisation of the musicians. The graphical contours are defined in JMSL as a series of closely-spaced noteheads that create connected line segments, open to broad interpretation by instrumentalists and to standard LOLC transformation operations by the laptop

Figure 3 SGLC Rehearsal with Sonic Generator

musicians. The textual instructions ask musicians to borrow and/or transform motives they or other musicians have played, mimicking the networked collaboration of the laptop musicians in in the acoustic realm.

The musical structure, presented in textual form to the perfomers, defines the piece as a progression of seven discrete sections, with each specifying a set of pre-composed fragments the laptop players may use and a set of operations they may use to transform them. In all sections, the laptop musicians are encouraged to borrow material created by other players. Generally speaking, the piece begins with a narrowly constrained set of unmodified long tones, then gradually allows additional pitch material and operations upon the material. Following that, the music transitions into the melodic motives, then gradually adds in transformative operations and the guided improvisations. The instrumental musicians respond to the notation, to instructions typed over the text chat, and to each other, musically interpreting the score and reacting to the graphical and textual notation to create a coherent, expressive musical performance.

SGLC was premiered by Sonic Generator, the professional contemporary chamber ensemble in residency at Georgia Tech, in February 2012(Figure 3).

6. CONCLUSION AND FUTURE WORK

The real-time notation extensions to LOLC described in this paper offer a novel approach to deeply integrating instrumental musicians into an improvising laptop ensemble, pushing both entities to the extremes of the excitement and risk of live performance: extreme improvisation through text-based collaboration and extreme sight-reading through real-time notation display. Yet LOLC is simultaneously designed to be accessible to laptop musicians, instrumental performers, and audiences: the command syntax and instant-messaging-style interface make LOLC easy for musicians with limited technical abilities to learn; the grounding of real-time notation in pre-composed fragments simplifies the daunting task of sight-reading; and the visualization of the performance, including text chat messages, reveals the complexity (and sometimes humor) of the collaboration to audiences.

We are currently conducting a formal evaluation of the real-time notation components of LOLC through quantitative analysis of server logs and qualitative interviews with performers, and we plan to make improvements to the system accordingly. This follows a similar approach we used to study the digital audio version of LOLC[14]. We also hope to further explore the use of LOLC as an educational tool to teach improvisation in the context of computing. Finally, we are looking forward to creating a series of compositions, both ourselves and in collaboration with other composers, that continue to explore the possibilities of LOLC's real-time notation and to push the boundaries of live performance.

7. ACKNOWLEDGMENTS

LOLC is supported by a grant from the National Science Foundation as part of a larger research project on musical improvisation in performance and education (NSF CreativeIT 0855758). In addition to the authors, Akito Van Troyer, Shannon Yao, Sidharth Subramanian, Scott McCoid, and Jung-Bin Yim have all made substantial contributions to designing and developing LOLC. We also extend our thanks to all the musicians of Sonic Generator. LOLC is available at http://www.jasonfreeman.net/lolc/. The SGLC score and performance video is available at http://www.jasonfreeman.net/sglc/.

8. REFERENCES

[1] Berliner, P. *Thinking in jazz: The infinite art of improvisation*. University of Chicago Press. 1994

[2] Brown, A.R. and Sorensen, A. Interacting with generative music through live coding. *Contemporary Music Review*, 2009. 28(1)17-29.

[3] Brown, C. and Bischoff, J. Indigenous to the Net: early network music bands in the San Francisco Bay area. 2002. Available online at http://crossfade.walkerart.org /brownbischoff/IndigenoustotheNetPrint.html.

[4] Brown, E. Calder Piece. The Earle Brown Music Foundation, New York, 1966.

[5] Collins, N.,McLean, A., Rohrhuber, J. and Ward, A. Live coding in laptop performance. *Organised Sound*, 2003. 8(3). 321-330.

[6] Didkovsky, N. and Burk, P. . Java Music Specification Language, an introduction and overview. In *Proceedings of International Computer Music*, San Francisco, California, 2001. 123-126

[7] Essl, K., Champ d'Action. Available online at www.essl.at/works/champ.html 1998.

[8] Freeman, J., Extreme sight-reading, mediated expression, and audience participation: Real-time music notation in live performance. Computer Music Journal, 2008. 32(3): p. 25-41.

[9] Freeman, J. and Colella, A. , Tools for Real-Time Music Notation. Contemporary Music Review, 2010. 29(1): p. 101-113.

[10] Freeman, J. and Godfrey,M. Creative collaboration between audiences and musicians in Flock. Digital Creativity, 2010. 21(2), 85-99.

[11] Freeman, J. and Van Troyer, A. Collaborative textual improvisation in a laptop ensemble. Computer Music Journal, 2011. 35(2). 8-21.

[12] Hiller, L. and Isaacson, L.M., Illiac suite, for string quartet. Vol. 30. 1957: New Music Edition.

[13] Kim-Boyle, D. Musical score generation in Valses and Etudes. In *Proceedings International Conference on New Interfaces for Musical Expression*, Vancouver, Canada, 2005. 238–239

[14] Lee, S.W., Freeman, J., Colella, A., Yao, S., and Van Troyer, A. Collaborative musical improvisation in a laptop ensemble with LOLC. In *Proceedings ACM Creativity and Cognition,* Atlanta, GA, 2011.

[15] Lewis, G., Virtual Concerto. Unpublished musical score, 2004.

[16] Rebelo, P., Notating the Unpredictable. Contemporary Music Review, 2010. 29(1): p. 17-27.

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

[18] Wang, G., Misra, A., Davidson, P., and Cook, P.R. CoAudicle: A Collaborative Audio Programming Space. in *Proceedings of the International Computer Music Conference*, Barcelona, Spain, 2005, 331– 334.

[19] Zorn, J. John Zorn's Cobra: Live at the Knitting Factory. New York: Knitting Factory Works KFW124, 1995. compact disc.