5. DISCUSSIONS

5.1 Improvement of Learning Experience

Our educational platform worked successfully in terms of the technical side throughout the workshop, but there were some points to be considered in relation to the participants’ experience. Although the students seemed to learn how to operate the application very quickly so that they actively designed their work by themselves, we wonder if the workshop covered all types of learning. Participants were able to submit materials from both within and outside the venue, where they were allowed to use their own smartphones to control objects on the screen in order to create a musical performance.

5.2 Future Works

For the next step of this project, we expect that students will accumulate knowledge about generative art by participating in workshops. For this purpose, it is necessary to conduct our activities in various places and to open more workshops. Our platform is designed so that the workshop system can function simultaneously in multiple locations. We are currently planning to deploy the workshop system as a package, so that any educational institution can easily hold workshops independently. Furthermore, we wish to allow participants to share their work online at any time.

Social media can also be utilized to develop an online posting forum to show the students’ works using MUCCA.

6. CONCLUSIONS

In this paper, we described the implementation and practice of our integrated educational platform, MUCCA. MUCCA is based on generative audio/visual art, and focuses on both the technical development and management of a workshop. The development of this platform was successful in terms of involving participants and their pieces; however, we need to further examine students’ learning experiences. We believe that MUCCA will become a sufficient platform for stimulating knowledge based on educational contents that help people use generative art to express themselves creatively.

Acknowledgments

The author would like to thank Masao Tsutsumi (a staff member of the Gifu city government), Yukiko Nishii (the accordion player), Jo Miura (the saxophone player), Sayumi Higo (the visual designer), and all other members who were involved in the workshop.

7. REFERENCES


Electro Contra: Innovation for Tradition

Benjamin D. Smith
Department of Music and Arts Technology, School of Engineering and Technology, Indiana University-Purdue University Indianapolis
bdsmith@iupui.edu

ABSTRACT

Technological interventions in American traditional fiddle and dance music are presented and specific design and development problems are considered. As folk dance communities and events explore the notion of incorporating modern electronic dance music into the experience certain inherent problems are exposed. Maintaining strict musical forms that are required for the traditional choreography, while maintaining the fluidity and control of live bands, and interacting with the other performers require new software tools. Initial solutions developed in Ableton Live are described and show a successful method of solving these challenges.

1. INTRODUCTION

Traditional aural music practices around the world evolve and maintain currency with the incorporation of new musical instruments and technologies. In the twentieth century, steel strings for guitars and violins, the advent of amplification and electric instruments, and increased manufacture and access to instruments had transformative impacts on music around the world. New genres grew out of the new technologies, such as Jazz and Rock and Roll, exploding in dance halls and on concert stages alike. Amplification is now a ubiquitous aspect of dance music performance in nearly every genre, from social and couples folk dancing to swing to electronic dance music (EDM). Today, computers present an immense domain of musical possibilities and their incorporation as a performance tool in traditional folk music, alongside fiddles and banjos, is already underway.

Performing traditional music electronically, on a technical level, presents many challenges to the electronic musician using currently available software tools. Most folk dance choreography fits strict musical forms and any musical deviations will disrupt the dancers and stop the dance. The music has to start and line up with the figures of the specific dance, requiring the musician to synchronize the choreography with the music. Further, the music is expected to dynamically respond to the dancers through texture changes and growth of a song, facilitating energetic and dramatic effects.

Based on these challenges several new software tools (plug-ins for Ableton Live Suite) have been designed.

Copyright © 2016 Benjamin D. Smith et al. This is an open-access article distributed under the terms of the Creative Commons Attribution License 3.0 Unported, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Electro Contra dance is a vibrant living tradition of dancing and music performance that has been steadily growing in popularity since the 1970s. Involving instruments, music, and choreography derived from eighteenth century practices in the British Isles, contra dance now has active communities across North America, Europe, and Australia. The current form of contra dance was first seen in the U.S. in the 1780s [3], and after disappearing from practice in the following century was reborn during the folk revival in the United States in the 1970s [8]. While the closely related forms of English, Scottish, and Irish dance followed the same trajectory they have become historically oriented practices, privileging traditional choreography and costumes. Uniquely, contra dance actively incorporates inter- and individual variation, new choreography, and experimentation with the forms and music [4].

The structure of contra dance employs two lines of dancers (the designation “contra” refers to this opposition of lines), who progressively move along the lines to dance with other individuals. The choreography typically involves a set of four dancers (two couples) executing a series of steps in unison that take up the 64 beats of the written dance [1]. All the dancers execute each figure in the dance concurrently and a series of 4-8 figures typically comprises a “dance,” which is then repeated 12-20 times along with live musical accompaniment.

The vast majority of the choreography is set to a binary musical form of AABH, wherein each section is 16 beats long. The music is performed live and is historically rooted in the traditional music of the British Isles (Irish and Scottish “fiddle” tunes). The meter is most commonly 2/2 or 6/8, and is strongly phrased to indicate the 8 bar sections, which dancers rely on for structural cues and to “keep them on track” [3]. Dance tempo does not vary widely, and is conventionally in the 115-125 beats-per-minute range [4].

The notion of “tradition” is integral to contemporary contra dancing, and the ideals of a non-commercial folk
The primary problems faced by live electronic music in the contra dance context stem from the strict requirements of the phrase structure and the need to aurally cue and indicate the repetitions in the form. The binary pattern of AABB, as well as the continual recycling of the whole form (over each 8-10 minute dance), are expected and relied on by the dancers [3]. This stands in contrast to the typical pop music song form of AABB and EDM forms which focus on continuity and minimal transience, like repetition. Further, pop songs commonly deviate from 32 bar forms to include a bridge section or other variations, which precludes their use in this context.

The electronic performer can create the form by discarding loops and playing everything live using controllers and MIDI interfaces (i.e. treating their setup like an acoustic instrument and ‘playing all the notes’). However this denies the hallmark sounds, sampled loops, and operating principles EDM is based on. The opposite approach seen above, of acoustic musicians applying pop song forms for dance, merely appropriates the content of one genre and transposes it to another, rather than exploiting the potential of fully blending the genres.

Groups preferring dance music must additionally be able to recover from errors enacted by the caller or dancers. While not common, either the caller may mistakenly call a figure or the dancer may forget and cause the dance to get out of sync with the music or come to a stop. It is imperative that the musicians are able to either re synchronize with the dance (by adding a few beats or skirting communities in the song), or quickly reset and recover by starting over.

An additional problem arises solely at the commencement of each dance where the musicians must either use the start of the choreography or align with the caller/or. Conventional acoustic contra dance bands start each dance in one of two ways: either by playing a short four beat introduction to indicate the start of the dance to the dancers, or by playing a repetitious musical pattern in the tempo of the dance and allowing the caller to time the figures to the music. In this latter case once the dancers are all in motion the musicians will seamlessly transition to their full tune/song/arrangement.

Ableton Live is a preferred software solution for many live electronic musicians playing on the contra dance stage due to its flexibility and interactivity (see Fig. 1). The ability to play loops, clips, and songs dynamically and apply further manipulations is the basis for these performances. However the challenges of phrase alignment in this environment are seen as cumbersome and constrain the expressive performance. For example, if the user wants to change material in the middle of the 32-bar form there is no easy way to quickly trigger new loops and cross-fade or cut the old ones while still ensuring adherence to the dance structure. If the user accidently triggers clips or sections at the wrong time there is no way to recover without impact to the musical form.

NEW DEVELOPMENTS

Based on discussions with performing musicians three Live “devices” (plug-ins or utilities in Live’s parlance) were proposed, developed, and tested. The overall goal is to ensure enforcement of the phrase structure, freeing the musician to focus on musical choices, texture and dynamic direction. The developed assistive utilities are:

1. **Sync**. A clip or section that transiently skips the entire session (all playing clips and events) to a specified bar and beat, or by a relative number of beats.

2. **Jump**. Jump device that skips a single track to a specified bar and beat, or by a relative number of beats.

3. **Clip Synchronization Device**. A clip synchronization device that maintains phrase alignment between a slave track and a master track (or the master clock).

All of these devices were built using Max For Live (M4L), working extensively through the Live API (in Max 7/2.2). This allowed easy modification during the prototyping stage as well as cross-platform distribution. These devices are currently scheduled for inclusion in Ableton Live, including generating bug lists and feature requests stemming from real-world application.

The jump song device (see Fig. 2) gives the player the ability to skip the song forward and backwards by single beats, assisting alignment with the dance if the music is out of sync, as well as jumping by whole sections to extend or shorten a song. This is analogous to a DJ moving the needle on a record, skipping the song to a new point in time. Ableton Live employs a model where each loop is essentially an individual record with its own needle, and jumping the song causes all the clips to jump synchronously. The Live API exposes access to the master clock time (“current_song_time”) which is set in the M4L device (through the “jump by” function) when the user enters a new absolute or relative jump time.

The track jump device performs similarly but only acts on a single track at a time, serving artistic effects and affording the alignment of different clips and loops. This uses the Live API “playing_position” property of a specific clip.

The clip synchronization device forces any track to stay aligned with either another track or the master clock. In Live the user can configure a quantization rate for clip launching, which can delay the clip to a slave track to match a certain phrase length. That is, if the quantization rate is set at 2 bars clips will start playing when the master song clock is at even bar numbers regardless of when the user presses the clip launch button (see Fig. 3, showing misaligning resulting from the user triggering clips around the phrase point). While this effectively enforces clip alignment dynamically, longer phrase lengths (such as the 8 or 32 bar phrases in contra dances) present challenges and this quantization limits performer spontaneity. If the user triggers a clip one beat after the bar quantization point the clip will wait 7 bars before playing (see Fig. 4). This limits the performer’s ability to improvisationally mix the music and dynamically trigger new clips. The new clip sync device allows the user to turn off the global quantization, allowing any clip to launch at any time, and the device ensures phrase alignment (see Fig. 5 where clips start launching in the middle of their loop). As each clip is launched the device skips it to the point that aligns it with the configured phrase length.

- **Figure 1** Ableton Live Set used for contra dances (image courtesy Julie Valimont) showing density of musical tracks and clips.
- **Figure 2** User interface for Jump device.
- **Figure 3** Loops with quantization at 2 bars, long loops enter out of phase with 8 bar phrases.
- **Figure 4** Loops with quantization at 8 bars, aligned correctly with phrases, but limited flexibility.
- **Figure 5** Loops with no quantization and Clip Sync device. Loops can start in the middle with guaranteed phrase alignment.
community and ‘traditional’ Americanisms are primary components in drawing many to the group [8]. As such these values are felt strongly amongst the community and guide many aspects of direction and organization locally and nationally. Musically, the ideals privilege ‘traditional’ folk acoustic instruments (such as the fiddle, pi- ano, banjo, and acoustic guitar), and tunes in strict musical forms (e.g. 2/2 metered Reels and Hoodoos; 6/8 metered Jigs and Marches).

However the authenticity of the ‘tradition’, in terms of longevity of customs and practices, is largely a chimera [8]. While some smaller communities in the North Eastern U.S. maintain a closer aural, generational link to the ancestral dance forms [9], for modern urban contra dance the authenticity of the musical tradition, in terms of reperto- rie and performance practices passed down aurally from generation to generation, is non-existent. The com- munity of dancers is international and associational, rather than based on ethnic, religious, or locational alignment [3].

The upholding of tradition creates friction with the liv- ing practice of contra dance, leading many con- temporary musical groups to both retain traditional in- strumentation while experimenting with a diversity of genres and sounds. One of the most popular national con- tra dance bands today, The Great Bear Trio [5], is lead by an electric guitar and regularly features arrangements of Top 40 radio songs. Another extremely popular band, Perpetual Motion, is an electric guitar and regularly features arrangements of Top 40 radio songs.

The first noted contra dancing to non- traditional pre-recorded music at mainstream contra dance events is thought to have occurred in the early 2000s in the Boston area [6]. This lead to alternative dan- ces colloquially termed “techno contra” [2], being staged across the U.S. today. Many self-styled DJs use contra dance colloquially termed “techno contras” [2], being the “build-up” and “drop” [7]. The receptiveness of the contra dance events appears to be based on fostering intense emotional experiences [7] and perceived “altered states”. This stands in contrast to the typical pop music song form of AABB and EDM forms which focus on continuity and minimalist trance like repetition. Further, pop songs commonly deviate from 32 bar forms to include a bridge section or other variations, which precludes the build-up.

The electronic performer can create the form by dis- carding loops and playing everything live using control- lers and MIDI interfaces (i.e. treating their setup like an acoustic instrument and ‘playing all the notes’). However this denies the hallmark sounds, sampled loops, and oper- ating principles EDM is based on. The opposite approach seen above, of acoustic musicians playing contemporary pop songs for dances, merely appropriates the content of one genre and transposes it to another, rather than ex- ploring the potential of fully blending the genres. Groups providing live music must additionally be able to recover from errors enacted by the caller or dancers. While not common, either the caller may mistakenly call a figure or the dancer may forget and cause the dance to get out of sync with the music or come to a stop. It is imperative that the musicians are able to either re synchronize with the dance (by adding a few beats or skipping over the song), or quickly reset and recover by starting over.

An additional problem arises solely at the commence- ment of each dance where the musician must either use the start of the choreography or align with the caller/leader. Conventional acoustic contra dance bands start each dance in one of two ways: either by playing a short four beat introduction to indicate the start of the dance to the dancers, or by playing a repetitious musical pattern in the tempo of the dance and allowing the caller to time the figures to the music. In this later case once the dancers are all in motion the music will seamlessly transition to their full tune/song/arrangement.

Ableton Live is a preferred software solution for many live electronic musicians playing on the contra dance stage due to its flexibility and interactivity (see Fig. 1). The ability to play loops, clips, and songs dynamically and apply further manipulations is the basis for the performance. However the challenges of phrase alignment in this environment are seen as cumbersome and constrain- ing to expressive performance. For example, if the user wants to change material in the middle of the 32-bar form there is no easy way to quickly trigger new loops and cross-fade or cut the old ones while still ensuring adher- ence to the dance structure. If the user accidentally triggers clips or sections at the wrong time there is no way to re- cover without impact to the musical form.

NEW DEVELOPMENTS

Based on discussions with performing musicians three Live “devices” (plug-ins or utilities in Live’s parlance) were proposed, developed, and tested. The overall goal is to ensure enforcement of the phrase structure, freeing the musician to focus on musical choices, texture and dynam- ic direction. The developed assistive utilities are:

1) Song jumping device that instantly skips the entire progression (all playing clips and events) to a specified bar and beat, or by a relative number of beats.
2) Track jump device that skips a single track to a specified bar and beat, or by a relative number of beats.
3) Clip synchronization device that maintains phrase alignment between a slave track and a master track (or the master clock).

All of these devices were built using Max For Live (M4L), working extensively through the Live API (in Max 7.2.2). This allowed easy modification during the prototyping stage as well as cross-platform distribution. These devices act on specific tracks during performance, generating bug lists and feature requests stemming from real-world application.

The song jumping device (see Fig. 2) gives the player the ability to skip the song forward and backwards by single beats, assisting alignment with the dance if the music is out of sync, as well as jumping by whole sec- tions to extend or shorten a song. This is analogous to a DJ moving the needle on a record, skipping the song to a new point in time. Ableton Live employs a model where each loop is essentially an individual record with its own needle, and jumping the song causes all the clips to jump synchronously. The Live API exposes access to the master clock time (“current_song_time”) which is set in the M4L device (through the “jump_by” function) when the user enters a new absolute or relative jump time.

The track jumping device performs similarly but only acts on a single track at a time, serving artistic effects and affording the alignment of different clips and loops. This uses the Live API “playing_position” property of a specific clip.

The clip synchronization device forces any track to stay aligned with either another track or the master clock. In Live the user can configure a quantization rate for clip launching, which causes the clip to either resynchronize with the dance (by adding a few beats or skipping over the song), or quickly reset and recover by starting over. When a clip is launched, it triggers a clip launch button (see Fig. 3, showing misalignment resulting from the user triggering clips around the phrase point). While this effectively enforces clip alignment dynamically, longer phrase lengths (such as the 8 or 32 bar phrases in contra dances) present chal- lenges and this quantization limits performer spontaneity. If the user triggers a clip one beat after the 8 bar quantiza- tion point the clip will wait 7 bars before playing (see Fig. 4). This limits the performer’s ability to improvisa- tionally mix the music and dynamically trigger new clips.

The new clip sync device allows the user to turn off the global quantization, allowing any clip to launch at any time, and the device ensures phrase alignment (see Fig. 5 where clips start playing in the middle of their loop). As each clip is launched the device skips it to the play point that aligns it with the configured phrase length.

Figure 1. Ableton Live Set used for contra dances (image courtesy: Julie Valmout) showing density of musical tracks and clips.

Figure 2. User interface for Song jump device.

Figure 3. Loops with quantization at 2 bars, long loops enter out of phase with 8 bar phrases.

Figure 4. Loops with quantization at 8 bars, aligned correctly with phrases, but limited flexibility.

Figure 5. Loops with no quantization and Clip Sync device. Loops can start in the middle with guaranteed phrase alignment.

In this device the phrase length can be set independently for each clip by the player (commonly 8 or 32 bars). For each given time point (t) the audio sample to play (Y(t)) is calculated from the time point of the master track (x(t)) and the phrase length (LP) by:

\[
Y(t) = \begin{cases} 
0 & \text{if } t \notin [t_n, t_{n+1}) \\
\frac{\text{LP}}{T} \cdot (t - t_n) & \text{if } t_{n+1} \leq t < t_{n+1} + \text{LP} \\
\end{cases}
\]
LR.step, an Algorithmic Drum Sequencer
Morgan Jenks
Texas A&M University
morgan.m.jenks@gmail.com

ABSTRACT
This paper presents a new algorithmic drum sequencer, LR.step. This sequence is based on Clarence Barlow’s Indispensability algorithm, and builds upon previous work with this algorithm, introducing several novel features. LR.step differs from previous implementations of the indispensability algorithm in that it features a method for calculating arbitrary subdivisions of the beat, such as 14/8 note triplets. This paper introduces two new processes for generating syncopation. Details of the software and possibilities for future work are given.

1. INTRODUCTION
Music software design has become widely accessible with the advent of the Internet and online communities of practice. Vast collective knowledge enables creators to design instruments that suit their own needs and ideals. LR.step exists as a result of a personal performance practice at the intersection of consumer electronics, longstanding research into music theory, the Max/MSP community, and my own stylistic interests, informed by ready access to various experimental beat makers such as Autechre[1].

1.1 The Algorithm
Many algorithmic sequencer techniques have been developed, including euclidean approaches, stochastic approaches, cellular automata, and genetic approaches [2,3]. Among all of these, the indispensability algorithm, developed by Clarence Barlow in 1978, stands out as an interesting balance of musicality and flexibility [4]. The indispensability algorithm, to summarize, sorts all inputs which I have named Irrationality and Eccentricity. LR.step is not stochastic. It is fully determinate and will work with this algorithm, introducing several novel features. LR.step differs from previous implementations of the indispensability algorithm in that it features a method for calculating arbitrary subdivisions of the beat, such as 14/8 note triplets. This paper introduces two new processes for generating syncopation. Details of the software and possibilities for future work are given.

2. SEQUENCER DETAILS
LR.step, in its current form, is a Max/MSP patch that syncs with Ableton Live via the ReWire protocol. In contrast to the Ableton and Kinetic rhythm generators, LR.step is not stochastic. It is fully determinate and will output a consistent and static pattern for a given combination of parameters. If indeterminate variations are desired, it may be mapped to any sort of modulator. Among the parameters, three stand out as novel developments: freely definable step sizes, and two syncopation parameters which have been named Irrationality and Eccentricity.

Copyright © 2016 First author et al. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY-Unported), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Figure 1. The indispensability set for one measure of 4/4 in 8th notes

The indispensability algorithm is a state machine much like cellular automata or euclidean rhythms. However whereas the latter two output sequences of Boolean values, indispensability sets provide a rich hierarchy for all possible pulses in a sequence and create conventional metrical emphasis, even in complex time signatures. The indispensability algorithm reveals connections between rhythm and harmony, and outputs patterns strikingly similar to traditional musics, for example, Franconian dance pieces [5]. Composer Georg Hajdu has ported the algorithm to Max/MSP and used it to assist in organizing a 19-tone equal temperament recorder piece among other things [5].

Barlow’s original implementation of the formula took the form of his all-in-one procedural constitution system, AutoBux[4]. Another implementation of the indispensability algorithm by Storos, Guedes and the Kinetic Controller Driven Adaptive Music Systems Project at the University of Texas at Austin, uses the indispensability set as a primary input in their control of meter, subdivisions, and probability weight [6]. As Elridge advocates regarding musical generativity, the indispensability algorithm is not lifted from another scientific context such as flocking simulation, but was devised specifically from harmonic and metrical principles [7].

ABSTRACT
This paper presents a new algorithmic drum sequencer, LR.step. This sequence is based on Clarence Barlow’s Indispensability algorithm, and builds upon previous work with this algorithm, introducing several novel features. LR.step differs from previous implementations of the indispensability algorithm in that it features a method for calculating arbitrary subdivisions of the beat, such as 14/8 note triplets. This paper introduces two new processes for generating syncopation. Details of the software and possibilities for future work are given.

1. INTRODUCTION
Music software design has become widely accessible with the advent of the Internet and online communities of practice. Vast collective knowledge enables creators to design instruments that suit their own needs and ideals. LR.step exists as a result of a personal performance practice at the intersection of consumer electronics, longstanding research into music theory, the Max/MSP community, and my own stylistic interests, informed by ready access to various experimental beat makers such as Autechre[1].

1.1 The Algorithm
Many algorithmic sequencer techniques have been developed, including euclidean approaches, stochastic approaches, cellular automata, and genetic approaches [2, 3]. Among all of these, the indispensability algorithm, developed by Clarence Barlow in 1978, stands out as an interesting balance of musicality and flexibility [4]. The indispensability algorithm, to summarize, sorts all inputs which I have named Irrationality and Eccentricity. LR.step is not stochastic. It is fully determinate and will work with this algorithm, introducing several novel features. LR.step differs from previous implementations of the indispensability algorithm in that it features a method for calculating arbitrary subdivisions of the beat, such as 14/8 note triplets. This paper introduces two new processes for generating syncopation. Details of the software and possibilities for future work are given.

2. SEQUENCER DETAILS
LR.step, in its current form, is a Max/MSP patch that syncs with Ableton Live via the ReWire protocol. In contrast to the AutoBux and Kinetic rhythm generators, LR.step is not stochastic. It is fully determinate and will output a consistent and static pattern for a given combination of parameters. If indeterminate variations are desired, it may be mapped to any sort of modulator. Among the parameters, three stand out as novel developments: freely definable step sizes, and two syncopation parameters which have been named Irrationality and Eccentricity.

Copyright © 2016 First author et al. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY-Unported), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Figure 1. The indispensability set for one measure of 4/4 in 8th notes

The indispensability algorithm is a state machine much like cellular automata or euclidean rhythms. However whereas the latter two output sequences of Boolean values, indispensability sets provide a rich hierarchy for all possible pulses in a sequence and create conventional metrical emphasis, even in complex time signatures. The indispensability algorithm reveals connections between rhythm and harmony, and outputs patterns strikingly similar to traditional musics, for example, Franconian dance pieces [5]. Composer Georg Hajdu has ported the algorithm to Max/MSP and used it to assist in organizing a 19-tone equal temperament recorder piece among other things [5].

Barlow’s original implementation of the formula took the form of his all-in-one procedural constitution system, AutoBux[4]. Another implementation of the indispensability algorithm by Storos, Guedes and the Kinetic Controller Driven Adaptive Music Systems Project at the University of Texas at Austin, uses the indispensability set as a primary input in their control of meter, subdivisions, and probability weight [6]. As Elridge advocates regarding musical generativity, the indispensability algorithm is not lifted from another scientific context such as flocking simulation, but was devised specifically from harmonic and metrical principles [7].

ABSTRACT
This paper presents a new algorithmic drum sequencer, LR.step. This sequence is based on Clarence Barlow’s Indispensability algorithm, and builds upon previous work with this algorithm, introducing several novel features. LR.step differs from previous implementations of the indispensability algorithm in that it features a method for calculating arbitrary subdivisions of the beat, such as 14/8 note triplets. This paper introduces two new processes for generating syncopation. Details of the software and possibilities for future work are given.

1. INTRODUCTION
Music software design has become widely accessible with the advent of the Internet and online communities of practice. Vast collective knowledge enables creators to design instruments that suit their own needs and ideals. LR.step exists as a result of a personal performance practice at the intersection of consumer electronics, longstanding research into music theory, the Max/MSP community, and my own stylistic interests, informed by ready access to various experimental beat makers such as Autechre[1].

1.1 The Algorithm
Many algorithmic sequencer techniques have been developed, including euclidean approaches, stochastic approaches, cellular automata, and genetic approaches [2, 3]. Among all of these, the indispensability algorithm, developed by Clarence Barlow in 1978, stands out as an interesting balance of musicality and flexibility [4]. The indispensability algorithm, to summarize, sorts all inputs which I have named Irrationality and Eccentricity. LR.step is not stochastic. It is fully determinate and will work with this algorithm, introducing several novel features. LR.step differs from previous implementations of the indispensability algorithm in that it features a method for calculating arbitrary subdivisions of the beat, such as 14/8 note triplets. This paper introduces two new processes for generating syncopation. Details of the software and possibilities for future work are given.

2. SEQUENCER DETAILS
LR.step, in its current form, is a Max/MSP patch that syncs with Ableton Live via the ReWire protocol. In contrast to the AutoBux and Kinetic rhythm generators, LR.step is not stochastic. It is fully determinate and will output a consistent and static pattern for a given combination of parameters. If indeterminate variations are desired, it may be mapped to any sort of modulator. Among the parameters, three stand out as novel developments: freely definable step sizes, and two syncopation parameters which have been named Irrationality and Eccentricity.

Copyright © 2016 First author et al. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY-Unported), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.