

# Generative Music for Live Musicians: *An Unnatural Selection*

Arne Eigenfeldt

School for the Contemporary Arts  
Simon Fraser University  
Vancouver, Canada  
[arne\\_e@sfu.ca](mailto:arne_e@sfu.ca)

## Abstract

*An Unnatural Selection* is a generative musical composition for conductor, eight live musicians, robotic percussion, and Disklavier. It was commissioned by Vancouver's Turning Point Ensemble, and premiered in May 2014. Music for its three movements is generated live: the melodic, harmonic, and rhythmic material is based upon analysis of supplied corpora. The traditionally notated music is displayed as a score for the conductor, and individual parts are sent to eight iPads for the musicians to sight-read. The entire system is autonomous (although it does reference a pre-made score), using evolutionary algorithms to develop musical material. Video of the performance is available online.<sup>1</sup> This paper describes the system used to create the work, and the heuristic decisions made in both the system design and the composition itself.

## Introduction

*An Unnatural Selection* can be classified as art as research: the author is a composer who has spent the previous thirty years coding software systems that are used as compositional assistants and/or partners. In the last ten years, these systems have explored greater autonomy, arguably creating computationally creative musical production systems that produce music that would be considered creative if the author had produced them independently.

Music has a long history of computational systems created by artist-programmers, in which many aspects of the musical creative process are automated (Chadabe 1980; Lewis 2000; Rowe 2004). Most of these systems have been idiosyncratic, non-idiomatic production systems specific to the artist's musical intention; however, some attempts have been made at evaluation (Eigenfeldt *et al.* 2012).

The author's own investigation into creative software have included multi-agent systems that emulate human improvisational practices (Eigenfeldt 2006), constrained Markov selection (Eigenfeldt and Pasquier 2010), and corpus-based recombination (Eigenfeldt 2012). All of these systems operate in real-time, in that they generate their output in performance using commercially available synthesizers, which, unfortunately, offer limited representa-

tions of their highly complex acoustic models (Risset and Matthews 1969, Grey and Moorer 1977)

In order to bypass these audio limitations, the author's more recent research investigates the potential for generating music directly for live performers (Eigenfeldt and Pasquier 2012b). Complex issues arise when generating music for humans, both in terms of software engineering – e.g. producing complex musical notation for individual performers – and human computer interaction: asking musicians to read music for the first time during the performance, without rehearsal, and without recourse to improvisation. See Eigenfeldt (2014) for a detailed discussion of these matters.

## Previous Work

*An Unnatural Selection* builds upon the work of others in several areas, including genetic algorithms, real-time notation, and generative music.

## Evolutionary Algorithms

Evolutionary computation has been used within music for over two decades in various ways. Todd and Werner (1999) provide a good overview of the earlier musical explorations using such approaches, while Miranda and Biles (2007) provide a more recent survey. Very few of these approaches have been compositional in nature; instead, their foci have tended to be studies, rather than the generation of complete musical compositions.

Several real-time applications of GAs have been used, including Weinberg *et al.* (2008), which selected individuals from an Interactive Genetic Algorithm (IGA) suitable for the immediate situation within a real-time improvisation. Another approach (Beys 2009) used a fitness function that sought either similar or contrasting individuals to an immediate situation within an improvisation.

Waschka (2007) used a GA to generate contemporary art music. His explanation of the relationship of time within music is fundamental to understanding the potential for evolutionary algorithms within art-music: “unlike material objects, including some works of art, music is time-based. The changes heard in a piece over its duration and how those changes are handled can be the most important aspect of a work.” Waschka's *GenDash* has several important attributes, a number of which are unusual: an individual is a measure of music; all individuals in all genera-

<sup>1</sup> <https://aeigenfeldt.wordpress.com/works/music-for-robots-and-humans/>

tions are performed; the fitness function is random, leading to random selection; the composer chooses the initial population. Of note is the second stated attribute, the result of which is that “the evolutionary process itself, not the result of a particular number of iterations, constituted the music”. Waschka provides some justifications for his heuristic choices, suggesting that while they may not be observed in real-world compositional process, they do provide musically useful results.

EAs have been used successfully in experimental music and improvisation for several years. In most cases, artists have been able to overcome the main difficulty in applying such techniques to music – namely the difficulty of formulating an effective aesthetic fitness function – through a variety of heuristic methods. One particularly attractive feature of EAs to composers relates to the notion of musical development – the evolution of musical ideas over time – and its relationship to biological evolution. As music is a time-based art, the presentation of successive generations – rather than only the final generation – allows for the aural exposition of evolving musical ideas.

### Real-time Notation

The prospect of generating real-time notation is an established area of musical research, and has been approached from a variety of viewpoints: see Hajdu and Didkovsky (2009) for a general overview. Freeman (2010) has approached it as an opportunity for new collaborative paradigms of musical creativity, while Gutknecht *et al.* (2005) explored its potential for controlled improvisation. Kim-Boyle (2006) investigated open-form scores, and McClelland and Acorn (2008) studied composer-performer interactions. However, the complexity of musical notation (Stone 1980), limited these efforts to graphic representations, rather than traditional western music notation that affords more precise and detailed directions to performers.

Hajdu’s Quintet.net (2005) was an initial implementation of MaxScore (Didkovsky and Hajdu 2009), a publically available software package for the generation of standard western musical notation, one that allows for complexities of notation on the level of offline notation programs. *An Unnatural Selection* uses MaxScore for the generation of the conductor’s score, which is then parsed to individual iPads and custom coded software.

### Production Systems versus Compositions

The creation of a production system for *An Unnatural Selection* was concurrent with the conceptualization of the composition itself, which is often the case in the author’s practice. The desired musical results are imagined through audition, and the software is coded with these results in mind. The attraction of generativity rests in the ability for a musical work to be actuated in varying forms while still retaining some form of overall artistic control.

The author has chosen to create composition-specific, rather than general purpose, systems for two reasons: previous experience has shown that general systems tend to become so complex with added features as to obfuscate

any purposeful artistic use, and secondly, specifically designed systems allow for a design with a singular artistic output in mind.

As a result, some modules within the system used in *An Unnatural Selection* are specific to that work; however, it also builds upon earlier work (Eigenfeldt and Pasquier 2010) as well as contributing to successive works. Specifically, the analysis engine and generation engine can be considered a free-standing system, which I refer to as *PAT* (Probabilities and Tendencies); the evolutionary aspects are specific to *An Unnatural Selection*.

### The GA and its role as “Development tool”

As already mentioned, the use of genetic algorithms – modified or otherwise – are attractive to composers interested in musical development. While this method of composition has its roots in the Germanic tradition of the 18<sup>th</sup> and 19<sup>th</sup> centuries, it remains cognitively useful, since it provides listeners with a method of understanding the unfolding of music over time (Deliege 1996; Deliege *et al.* 1996). A description of the work from the program notes – “musical ideas are born, are passed on to new generations and evolved, and eventually die out, replaced by new ideas” – may suggest principles of artificial life, or music based upon Brahms, Mahler, or Schoenberg.

A general conception of the first movement was a progression from chaos to order;

- an initial population of eight musical phrases are presented concurrently by the eight instrumentalists;
- the phrases are repeated, and each repetition develops the phrases independently;
- segments from the individual phrases infiltrate one another;
- the individual phrases separate in time, thus allowing their clearer perception by the listener.

While these concepts began with a musical aesthetic in mind, they were clearly influenced by their potential inclusion of genetic algorithms.

### The Score as template

*An Unnatural Selection* is the most developed system in my pursuit of real-time composition (Eigenfeldt 2011); the possibility to control multiple complex gestures during performance. As will be described, *An Unnatural Selection* involves a number of high-level parameter variables that determines how the system generates and evolves individuals; dynamically controlling these in performance effectively shapes the music. As the performance approached, I doubted my performative abilities, and instantiated a score-based system that allowed for the pre-determined setting of the control parameters for each successive generation: while the details of work would still be left to the system, the overall shape would be preset. The use of such templates is not uncommon in other computationally creative media: Colton *et al.* (2012) used similar design constraints in generating poetry in order to maintain formal patterns.

## Probabilities and Tendencies (PAT)

The heart of *PAT* rests in its ability to derive generative rules through the analysis of supplied corpora. Cope (1987) was the first composer to investigate the potential for style modeling within music; his Experiments in Musical Intelligence generated many compositions in the style of Bach, Mozart, Gershwin, and Cope. Dubnov *et al.* (2003) suggest that statistical approaches to style modeling “capture some of the statistical redundancies without explicitly modeling the higher-level abstractions”, which allow for the possibility of generating “new instances of musical sequences that reflect an explicit musical style”. However, their goals were more general in that composition was only one of many possible suggested outcomes from their initial work. Dubnov’s later work has focused upon machine improvisation (Assayag *et al.* 2010).

The concept of style extraction for reasons other than artistic creation has been researched more recently by Collins (2011), who tentatively suggested that, given the state of current research, it *may* be possible to successfully generate compositions within a style, given an existing database.

For *An Unnatural Selection*, corpora included compositions by the following composers:

- Movement I: 19 compositions by Pat Metheny
- Movement II: 2 compositions by Pat Metheny and 2 by Arvo Pärt
- Movement III: 1 composition by Terry Riley and 2 by Pat Metheny

These specific selections were arrived at through trial and error, as well as aesthetic consideration. The contemporary jazz material of Metheny provided harmonic richness without the functional tonality of the 19<sup>th</sup> century. Combining this corpus with Pärt’s simpler harmonies and melodies gave them an interesting new dimension, while the repetitive melodic material of Riley’s *In C*, when combined with Metheny’s harmonies created a new interpretation of minimalist melodic and rhythmic repetition with more complex harmonic underpinnings.

### Analysis of corpora

*PAT* requires specially prepared MIDI files that consist of a quantized monophonic melody in one channel, and quantized harmonic data in another: essentially, a lead-sheet representation of the music. Prior to the creation of melodic, harmonic, and rhythmic n-gram dictionaries (Pearce and Wiggins 2004), harmonic data is parsed into pitch-class sets (Forte 1973). Melodic data is stored in reference to the harmonic set within which it was found, both as an actual MIDI note number and pitch-class as relative to the set.

### Representation

Music representation, and its problematic nature, has been thoroughly researched: Dannenburg (1993) gives an excellent overview of the issues involved. Event-lists are the standard method of symbolic representation currently used, as they supply the minimally required information for rep-

resenting music within a note-based paradigm. However, since event-lists do not capture relationships between events, they have proven problematic for generative purposes (Maxwell 2014). For this reason, *PAT* includes non-events that are displayed in music notation.



**Figure 1.** A notated melodic phrase, with beats 1 through 4 indicated, and non-events marked below.

Figure 1 presents a simple melodic phrase, and its event-based representation is shown in Table 1. While the onset times and durations are captured, their interrelationships, clearly shown in Figure 1, are difficult to determine. The initial event’s prolongation into the second beat, as shown through the tie (marked with an x), is missing. Similarly, the rest on the third beat (also marked with an x) segments the second and third beats, also not obvious in Table 1.

Event #	Beat	Pitch	Duration
1	1.0	60	1.5
2	2.5	62	0.5
3	3.5	64	0.5
4	4.0	65	1.0

**Table 1.** The music of Fig. 1, represented as events.

The solution in *PAT* is to include all non-events: rests are represented as pitch 0 with appropriate durations, and ties are represented as incoming pitches with negative durations: see Table 2.

Event	Beat	Pitch	Duration
1	1.0	60	1.5
2	2.0	60	-0.5
3	2.5	62	0.5
4	3.0	0	0.5
5	3.5	64	0.5
6	4.0	65	1.0

**Table 2.** The music of Fig. 1, showing the “non-events” 2 and 4.

Associations between events are retained within *PAT* through encoding by beat. As the generative engine uses Markov chains, the important relationships within and between beats are preserved through separate pitch and rhythm/duration n-gram dictionaries.

**Rhythm** Events are stored as onset/duration duples, grouped into beats, with onset times indicating offset into the beat. Thus, Figure 1, segmented into individual beats, is initially represented as:

```
(0.0 1.5)
(0.0 -0.5) (0.5 0.5)
(0.0 0.5) (0.5 0.5)
(0.0 1.0)
```

Each beat, as a duple or combination of duples, serves as an index into the rhythm n-gram dictionary, which stores all continuations and the number of times a continuation has been found. Thus, after encoding only Figure 1, the rhythm dictionary would consist of the following:

```
(0.0 1.5)
(0.0 -0.5) (0.5 0.5) 1
(0.0 -0.5)(0.5 0.5)
(0.0 0.5)(0.5 0.5) 1
(0.0 0.5)(0.5 0.5)
(0.0 1.0) 1
```

**Pitch** Melodic events are stored in relation to the harmonic set within which they occurred. The total number of occurrences of each pitch-class (PC), relative to the set, are stored, as well as PCs that are determined to begin phrases (initial PCs) and end phrases (terminal PCs). Lastly, an n-gram for the continuation of each PC (n>) is stored, along with an n-gram of its originating PC (>n).

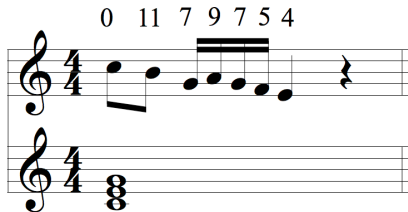


Figure 2. A melodic phrase with accompanying harmony; pitch-classes are indicated.

Thus, given the melodic and harmonic material of Figure 2, the melodic dictionary shown in Table 3 is constructed. Note that separate contour arrays are kept so as to retain actual melodic shapes.

Set:	0	4	7									
Pitch Class	0	1	2	3	4	5	6	7	8	9	10	11
Total PCs:	1	0	0	0	1	1	0	2	0	1	0	1
Initial:	1	0	0	0	0	0	0	0	0	0	0	0
Terminal:	0	0	0	0	1	0	0	0	0	0	0	0
0>	0	0	0	0	0	0	0	0	0	0	0	1
5>	0	0	0	0	1	0	0	0	0	0	0	0
7>	0	0	0	0	0	1	0	0	0	1	0	0
9>	0	0	0	0	0	0	0	1	0	0	0	0
11>	0	0	0	0	0	0	0	1	0	0	0	0
>4	0	0	0	0	0	1	0	0	0	0	0	0
>5	0	0	0	0	0	0	0	1	0	0	0	0
>7	0	0	0	0	0	0	0	0	0	1	0	1
>9	0	0	0	0	0	0	0	1	0	0	0	0
>11	1	0	0	0	0	0	0	0	0	0	0	0

Table 3. The music of Fig. 2, storing individual PC's movement to (n>) and from (>n), as well as a count of overall PCs for the set, and which PCs initiated and terminated phrases.

A similar system is used for harmony, with the n-gram storing the relative root movement of each set. Lastly, as well as melodic contours, an array of root movements (bass lines) is also kept. In both cases, these contours are normal-

ized and their length's scaled. New contours are compared to those existing using a Euclidean distance function, and those below a user-set minimum similarity level are culled, in order to avoid excessive similarity.

## Generation

The generative and evolutionary algorithms within *An Unnatural Selection* utilize user-set parameters that define how the algorithms function; it is the dynamic control of these parameters over time that shapes the music. As has been mentioned, *An Unnatural Selection* employs a parameter score to control these values.

### Evolutionary Methods in *An Unnatural Selection*

*An Unnatural Selection* uses the architecture of *PAT* within a modified evolutionary system. Within this system, musical phrases operate as individuals, or phenotypes, and individual beats – a combination of rhythmic and melodic material – operate as chromosomes. Phrases are developed in such ways that they represent successive generations. Since all individuals pass to the next generation, there is no selection, and thus no fitness function; however, each individual experiences significant crossover and mutation. Several independent populations exist simultaneously.

The use of evolutionary methods are extremely heuristic; earlier uses of such techniques by the author are documented elsewhere (Eigenfeldt 2012; Eigenfeldt and Pasquier 2012a).

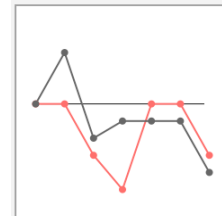


Figure 3. A root progression request (red), and the generated progression based upon possible continuations (grey).

### Generating Harmonic Progressions

A harmonic progression is the first generated element. A root progression is selected from the database as a target, and scaled by the requested number of chords in the progression. An initial chord is then selected from those sets that initiated phrases, and its continuations are compared to the next interval in the target. A Gaussian selection is then made from the highest probabilities. This process continues until a phrase progression is generated (see Figure 3). At this point, the progression has not been assigned individual durations.

### Generating Phrases/Individuals

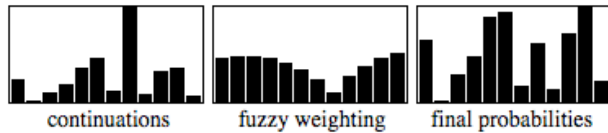
A number of required parameter values are calculated through a combination of corpus data and user-set ranges. For example, in order to select a phrase length for an individual, the actual phrase lengths from the corpus are ordered, and a value is sampled from this list from within a

user-set range (in this case *phraseLengthRange*). Thus, if this range is fixed between 0.9 and 1.0, a random selection will be made from 10% of the corpus' longest phrase lengths.

Individual phrases are assigned to specific instruments; since *An Unnatural Selection* was composed for eight instrument, Disklavier, and robotic percussion, the population consisted of a maximum of 12 individuals (the piano and percussion used two independent phrases). An important user parameter is whether the instrument (and thus the phrase) is considered *foreground* or *background*: in the case of the former, rhythmic data is selected from the corpus based upon density, while in the latter, data is selected based upon complexity (syncopation). Foreground individuals are deemed to be more active and have more variation; background individuals are either more repetitive or of longer duration, as set by a user parameter.

**Foreground** The number of onsets per beat is determined by a user parameter, *phraseDensityRange*. At initialization, the corpus' average beat density is scaled between 0.0 (the least dense) to 1.0 (the most dense), and a selection is made within the user range.

**Background** At initialization, the corpus' onsets are also rated for complexity: the relative amount of syncopation within each beat. Background phrases are comprised of either rhythmic material or held notes; in the case of the former, an exponential selection is made from the top 1/3 of the corpora (the most syncopated), while a similar selection is made from the bottom 1/3 for held individuals. Background phrases are immediately repeated if they are less than one measure in total duration.



**Figure 4.** The continuations for a specific PC (7), left; a weighting that favors more distant PCs, center; the final probability for PC selection, right.

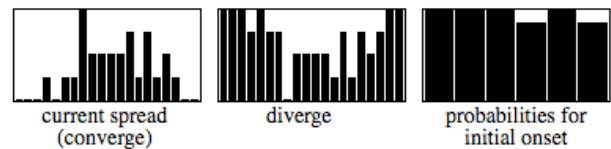
Once an initial selection is made for foreground or background individuals, the continuations from that beat are constrained by the same user parameters.

**Melodic material** Similar to harmonic and rhythmic generation, melodic generation selects an initial PC from those PCs in the corpus that began melodic phrases; continuations of that PC are then weighted to derive the probabilities for the next PC. In the case of foreground individuals, a fuzzy weighting is applied so as to avoid direct repetition and small intervals. (see Figure 4); for background phrases,

the opposite weighting is applied to avoid large melodic leaps.

**Individual locations within overall phrase** Once all phrases have been generated, the maximum length is determined, in beats; this value is rounded up to the next measure, and becomes the overall phrase length to which the harmonic progression is overlaid.

Individuals are placed within the overall phrase, either attempting to converge upon other individual's locations, or diverge from them, depending upon a user-set parameter *phraseVersusPhrase*. Each phrase's current onset locations are summed, which will determine the probability for the placement of individuals in the next overall phrase while the inverse will provide probabilities for divergence (see Figure 5). Rests are added to the beginning and/or end of the individual in order to place them in the overall phrase: these rests are not considered part of the individual.

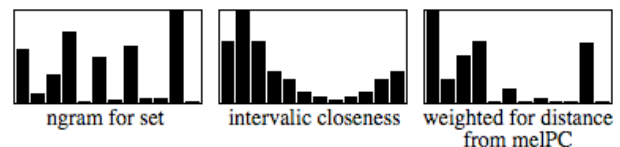


**Figure 5.** The number of total onsets per beat, left; the inverse as avoidance probability, center; the final probability for phrase starts, right. Because of the individual's length, its placement is limited to the first six locations of the overall phrase.

### Melodic Quantization

With the harmony now in place, PCs are quantized to sets within which they are located. A PC is compared to the total n-gram for its current harmonic set, which acts as an overall probability function, scaled by intervallic closeness to the PC (see Figure 6). In this way, PCs are not forced to a pre-defined "chord-scale" for the set, but adjusted to fit the n-gram for the set within the corpus.

Pitch ranges are then adjusted for each individual, and dynamics, articulations, slurs, and special text (i.e. *arco* vs. *pizzicato*) are applied: space does not allow for a discussion of how these parameters are determined.



**Figure 6.** The n-gram for the set (0 3 7 10), left; a weighting for a raw PC (1) that favors intervallic closeness; the final probability for PC quantization, right.

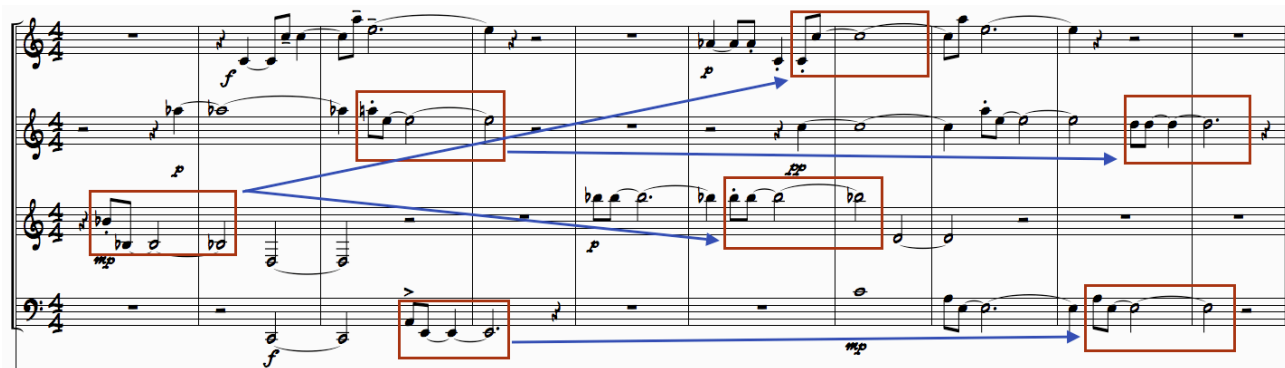


Figure 7. The first two generations of a population of four individuals, demonstrating crossover by segment.

## Evolving Populations

As mentioned previously, all individuals progress to the next generation, unless they are turned off in the user score. Evolution of individuals includes crossover (within set populations) and mutation.

**Crossover** The individual’s chromosomes are its beats; as rests are considered events within *PAT*, every beat, including rests, constitutes a separate chromosome. Crossover does not involve the usual splicing of two individuals, but instead the insertion or deletion of *musical segments* between individuals. Segmentation is done using standard perceptual cues, including pitch leaps, rests, and held notes (Cambouropoulos 2001), resulting in segments of one to several beats (see Figure 7).

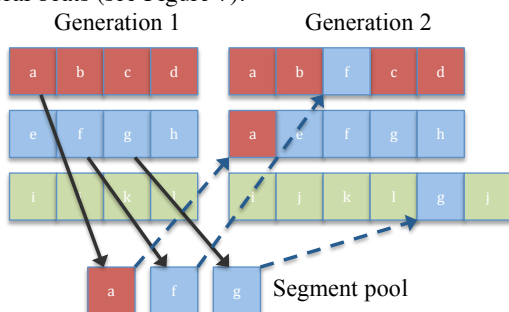


Figure 8. Two generations of three individuals (red, blue, green), showing expansion through crossover of segments. Segments *a*, *f*, and *g* are copied to the segment pool, potentially mutated, then inserted into other individuals

Individuals will either expand or contract during crossover, depending upon a user-set parameter. Contracting an individual involves deleting a segment, and splicing together the remaining parts in a musically intelligent way. Expansion involves copying segments from different individuals into a separate pool that contains a maximum of 16 segments, differentiated by individual type: foreground versus background (see Figure 8). Segments are potentially mutated (see next section), then inserted into individuals.

**Mutation** Mutation can occur on segments within the segment pool prior to insertion, or on the entire individual,

depending upon the user-set parameter *multiBeatProbability*. Mutations are musically useful variations, including:

- scramble – randomly scramble the pitch-classes;
- transpose – transpose a segment up or down by a fixed amount, from 2 pitch-classes to 12;
- sort+ - sort the pitch-classes from lowest to highest;
- sort- - sort the pitch-classes from highest to lowest;
- rest for notes – substitute rests for pitch-classes, to a maximum of 50% of the onsets in the segment.

The type of mutation is selected using a roulette-wheel selection method from user-set probability weightings for each type.

## Logistics

*An Unnatural Selection* is coded in MaxMSP<sup>2</sup>, using MaxScore for notational display. Custom software was written to display individual parts on iPads, which received JMSL (Didkovsky and Burke 2001) data wirelessly over a TCP network. The generative software composes several phrases in advance, and sends the MIDI data to Ableton Live<sup>3</sup> for performance (specifically the Disklavier and robotic percussion); Ableton Live provides a click track for the conductor, and sends messages back to the generative system requesting new material.

## Discussion

*An Unnatural Selection* is, first and foremost, an artistic system designed to create multiple versions of a specific composition – the author’s interpretation of “generative music”. Many aspects of the system’s development – for example, the multiple populations – were arrived at through artistic reasons, rather than scientific. Algorithms were adjusted and parameters “tweaked” through many hours of listening to the system’s output; as a result, heuristics form an important aspect of the final software.

Whether the system is computationally creative is a more difficult matter to determine. While I echo Cope’s desire that “what matters most is the music” (Cope 2005), I am fully aware of Wiggins reservations that “with hand-coded rules of whatever kind, we can never get away from

<sup>2</sup> [www.cycling74.com/](http://www.cycling74.com/)

<sup>3</sup> [www.ableton.com/](http://www.ableton.com/)

the claim that the creativity is coming from the programmer and not the program” (Wiggins 2008).

The overriding design aspect entailed musical production rules derived through analysis of a corpus; however, as I discuss elsewhere (Eigenfeldt 2013), how this data is interpreted is itself a heuristic decision, especially when being used to create an artwork of any value.

## Evaluation

While the intention of *An Unnatural Selection* was primarily artistic, the notion of evaluation was not entirely ignored, an issue the author has attempted to broach previously (Eigenfeldt *et al.* 2012). The work was clearly experimental: it would have been much easier to generate the music offline, and select the best examples of the system, allowing the musicians to rehearse and perform these in ways in which they are accustomed. However, the fact that the music was generated live was an integral element to the performance: in fact, interactive lighting was used in which the musician’s chairs were lit only while they played, an effort to underline the real-time aspect.

While no formal evaluation studies were done, the musicians were asked to critically comment upon their experiences. Their comments are summarized here.

**Limited Complexity in Structure** Some musicians commented on the relatively unsophisticated nature of the overall form of the generated music:

*“I didn’t sense a strong structural aspect to the pieces. I thought the program generated some interesting ideas but I would like to see more juxtaposition, contrast of elements, in order to create more variety and interest.”*

*“I would venture to say... that the music... certainly wasn’t as developed or thoughtful as something that a seasoned, professional composer would create.”*

*“...any of the versions would likely have struck me as somewhat interesting but fairly basic.”*

Generating convincing structure is an open problem in musical metacreation, which is not surprising, as it is one of the most difficult elements to teach young composers.

**More Overall Complexity** When asked for specific suggestions, several musicians provided very musical suggestions, including a greater variety of time signatures, more subtle instrumentation and playing techniques, different groupings of musicians, *accelerando* and *rubato*. Many of these aspects can, and will be incorporated into future versions of the system.

**Positive comments** Keeping in mind that these are professional musicians specializing in contemporary music performance, I was happy to receive positive comments:

*“I assume the software is going to continue to grow and become more accomplished through further exposure to, and analysis of, sophisticated compositional ideas.”*

*“I thought some of music was beautiful, especially in the second movement.”*

*“It seems to me that what you are doing is groundbreaking and interesting, even if still at a relatively primitive stage.”*

## Conclusion

*An Unnatural Selection* was the culmination of my research into generating music in real-time for live musicians. Upon reflection after the fact, my goal was to present musical notation to the performers that was as close as possible to what they were used to, since no improvisation would be expected. Naturally, this would necessitate having the musicians perform the music without any rehearsal – and extremely demanding request. While the extended rehearsals did allow the musicians to gain some expectations of what to expect from the software, it failed to provide them with what rehearsals usually provide: a time to discover the required interactions inherent within the music. One musician suggested that these indications, normally learned during rehearsal periods, could somehow appear in the notation:

*“Maybe the screen could indicate to the players when they have an important theme to bring out, and also indicate which instrument they are in a dialogue with or have the same rhythmic figure as?”*

Future versions of the system will explore this new paradigm, which also suggests the potential to involve the performers within the generative composition in ways that would not be possible without intelligent technology.

## Acknowledgements

This research was undertaken through a Research/Creation grant from Canada’s Social Science and Humanities Research Council (SSHRC). Thanks to the Generative Media Research Group as well as the Metacreation Lab for their support and suggestions during the creation of this work. Particular thanks goes to James Maxwell for his thought-provoking work on cognitive music generative systems.

## References

- Assayag, G., Bloch, G., Cont, A., & Dubnov, S. 2010. Interaction with Machine Improvisation. *The Structure of Style*, 219–245.
- Beys, P. 2009. Interactive Composing as the Expression of Autonomous Machine Motivations. *International Computer Music Conference (ICMC)*, Montreal, 267–74.
- Chadabe, J. 1980. Solo: A Specific Example of Realtime Performance. *Computer Music-Report on an International Project. Canadian Commission for UNESCO*.
- Cambouropoulos, E. 2001. Melodic cue abstraction, similarity, and category formation: A formal model. *Music Perception*, 18(3), 347–370.
- Collins, T. 2011. “Improved methods for pattern discovery in music, with applications in automated stylistic composition,” *PhD thesis*, Faculty of Mathematics, Computing and Technology, The Open University.

- Colton, S., Goodwin, J., & Veale, T. 2012. Full face poetry generation. *International Conference on Computational Creativity (ICCC)*, Dublin, 95–10.
- Cope, D. 1987 An Expert System for Computer-Assisted Composition. *Computer Music Journal*, 11(4), 30–46.
- Cope, D. 2005. *Computer models of musical creativity*, Cambridge: MIT Press.
- Dannenberg, R. 1993. Music representation issues, techniques, and systems. *Computer Music Journal*, 17(3), 20–30.
- Deliege, I. 1996. Cue abstraction as a component of categorisation processes in music listening. *Psychology of Music*, 24(2), 131–156.
- Deliège, I., Mélen, M., Stammers, D., and Cross, I. 1996. Musical schemata in real-time listening to a piece of music. *Music Perception*, 117–159.
- Didkovsky, N., Burk, P. 2001. Java Music Specification Language, an introduction and overview. *ICMC*, Havana, 123–126.
- Dubnov, S., Assayag, G., Lartillot, O., & Bejerano, G. 2003. “Using machine-learning methods for musical style modeling,” *Computer*, 36(10), 73–80.
- Eigenfeldt, A. 2006. Kinetic Engine: toward an intelligent improvising instrument. *Sound and Music Computing Conference (SMC)*, Marseilles, 97–100.
- Eigenfeldt, A., Pasquier, P. 2010. Realtime generation of harmonic progressions using controlled Markov selection. *ICCC*, Lisbon, 16–25.
- Eigenfeldt, A. 2011. Real-time composition as performance ecosystem. *Organised Sound*, 16(02), 145–153.
- Eigenfeldt, A. 2012. Corpus-based recombinant composition using a genetic algorithm. *Soft Computing*, 16(12), 2049–2056.
- Eigenfeldt, A., Pasquier, P. 2012a. Populations of Populations - Composing with Multiple Evolutionary Algorithms, P. Machado, J. Romero, and A. Carballal (Eds.). *EvoMUSART 2012*, LNCS 7247, 72–83.
- Eigenfeldt, A., Pasquier, P. 2012b. Creative Agents, Curatorial Agents, and Human-Agent Interaction in Coming Together. *SMC*, Copenhagen, 181–186.
- Eigenfeldt, A., Burnett, A., & Pasquier, P. 2012. Evaluating musical metacreation in a live performance context. *ICCC*, Dublin, 140–144.
- Eigenfeldt, A. 2013. The Human Fingerprint in Machine-Generated Music. *xCoAx: Computation, Communication, Aesthetics, and X*, Bergamo, 107–115
- Eigenfeldt, A. 2014. Generative Music for Live Performance: Experiences with real-time notation. *Organised Sound*, 19(3).
- Forte, A. 1973. *The structure of atonal music*. Yale University Press.
- Freeman, J. 2010. Web-based collaboration, live musical performance and open-form scores. *International Journal of Performance Arts & Digital Media*, 6(2), 149–170.
- Grey, J., Moorer, J. 1977. Perceptual evaluations of synthesized musical instrument tones. *The Journal of the Acoustical Society of America*, 62(2), 454–462.
- Gutknecht, J., Clay, A., & Frey, T. 2005. GoingPublik: using realtime global score synthesis. *New Interfaces for Musical Expression*, Singapore, 148–151.
- Hajdu, G. 2005. Quintet.net: An environment for composing and performing music on the internet. *Leonardo*, 38(1), 23–30.
- Hajdu, G., Didkovsky, N. 2009. On the Evolution of Music Notation in Network Music Environments, *Contemporary Music Review*, 28(4-5), 395–407.
- Kim-Boyle, D. 2006. Real time generation of open form scores. *Proceedings of Digital Art Weeks*, ETH Zurich.
- Lewis, G. 2000. Too many notes: Computers, complexity and culture in voyager. *Leonardo Music Journal*, 10, 33–39.
- McClelland, C., Alcorn, M. 2008. Exploring new composer/performer interactions using real-time notation. *ICMC*, Belfast, 176–179.
- Maxwell, J. 2014. Generative Music, Cognitive Modelling, and Computer-Assisted Composition in MusiCog and ManuScore. *PhD Thesis*, Simon Fraser University.
- Miranda, E., Biles, J. eds. 2007. *Evolutionary Computer Music*. London: Springer.
- Pearce, M., Wiggins, G. 2004. Improved methods for statistical modelling of monophonic music. *Journal of New Music Research*, 33(4), 367–385.
- Risset, J., Mathews, M. 1969. Analysis of musical instrument tones. *Physics today*, 22(2), 23–30.
- Rowe, R. 2004. *Machine musicianship*. MIT press.
- Stone, K. 1980. *Music Notation in the Twentieth Century*. W.W. Norton, New York.
- Todd, P., Werner, G. 1999. Frankensteinian methods for evolutionary music composition. *Musical networks: Parallel distributed perception and performance*, Griffith, N., Todd, P., eds., Cambridge, MA, 313–339.
- Waschka, R. 2007. Composing with Genetic Algorithms: GenDash. *Evolutionary Computer Music*, Springer, London, 117–136.
- Weinberg, G., Godfrey, M., Rae, A., & Rhoads, J. 2008. A Real-Time Genetic Algorithm in Human-Robot Musical Improvisation. *Computer Music Modeling and Retrieval, Sense of Sounds*, Springer, Berlin, 351–359.
- Wiggins, G. 2008. Computer Models of Musical Creativity: A Review of Computer Models of Musical Creativity by David Cope, *Literary and Linguistic Computing*, 10(1), 109–116.