Generating Structure –
Towards Large-Scale Formal Generation

Arne Eigenfeldt
School for the Contemporary Arts, Simon Fraser University
arne_e@sfu.ca

Abstract

Metacreative systems have been very successful at generating event-to-event musical details and generating short forms. However, large-scale formal decisions have tended to remain in the hands of a human, either through an operator guiding an improvisational system along in performance, or the assemblage of shorter generated sections into longer forms. This paper will describe the open problem of large-scale generative form, and the author's attempts at delegating such decisions to a metacreative system by describing several of his generative systems and their approach to structure. The author will describe in greater detail his latest system – *The Indifference Engine* – and how it negotiates between agent intentions and performance information derived from a live performer.

Introduction

Metacreation is the idea of endowing machines with creative behavior (Whitelaw 2004). It is a contemporary approach to generative art, which itself refers to any practice – computational or otherwise – where the artist creates a process that is set in motion with some degree of autonomy (after Galanter 2003). Metacreation employs tools and techniques from artificial intelligence, artificial life, and machine learning to develop software that exhibits behaviors that would be considered creative if performed by humans (Eigenfeldt and Pasquier 2013). In a field populated by both scientists and artists, it is meaningful that its results can be measured by the quality of the artistic output created by its practice-based researchers; while a significant amount of research within the field is in the form of scientific papers – test cases, studies, position papers, attempts at producing formula to define creativity – many artists in the field are setting as a priority the production of quality artwork. As such, artistic decisions are often considered paramount to pure scientific results.

This paper will present the author's experience with generating structure within generative music, beginning with a description of musical form, followed by a brief discussion of how form has been handled in generative and interactive musical systems. The author will then discuss his own work in using generative methods to create formal structures, and finally present a detailed description of his most recent system, *The Indifference Engine*, which is a multi-agent real-time system in which agents form beliefs and intentions, and alter their long range goals based both upon their understanding of the performance environment (i.e. a live performer), and negotiations carried out between the agents themselves.

On Musical Form

One of the most difficult aspects of musical composition – at least within Western art music – is the creation of form. In teaching both acoustic and electroacoustic composition to students for twenty years, the author has found that they tend to have minimal difficulty in coming up with ideas; however, shaping them over time, and creating a cohesive whole is a skill that takes years of practice to develop. Asking "what comes next?" is an incredibly difficult decision for young composers to answer with any certainty.

Traditionally, architectural forms that exist outside of the music – i.e. the sonata form – have allowed composers to pour music into an existing structure. Such forms were initially adopted by modernist composers such as John Cage through his use of durational frames (Pritchett 1996); Pierre Boulez, however, eloquently argues against this, and for the potential of deriving form from the material itself (Boulez et al. 1964), a process suggestive of evolutionary tendencies.

Such forms can be found within improvised music, in which performers react to one another's playing instantaneously during performance, and the resulting overall structure emerges within the performance itself. It is per-

---

haps not surprising that the vast majority of early metacreative systems have been based upon an improvisational model, shortly to be discussed in greater detail.

While musical organization can be discussed from a perceptual perspective involving expectation and emotion (Meyer 1956, Huron 2006, Temperley 2001), this research has tended to be limited to short time scales rather than overall structure. Other approaches to understanding musical form consider phrasal structures (Narmour 1999, Negretto 2012), as well as top-down approaches (Hindemith 1970, Schenker 1972, Lerdahl and Jackendoff 1983).

While the vast majority of the literature on computational creativity is looking at composition and often reaches a human-competitive level for short excerpts, long-term dependencies and overall structure are harder to capture. Furthermore, as noted in the reflections on the first Musical Metacreation Festival (Eigenfeldt et al. 2013), the delegation of large-scale musical structure to a system was questioned as even being artistically desirable: many system designers felt that they wanted to remain “in the loop”, and thus control form interactively.

Previous Work

Within systems that generate form, the two approaches used by human composers have also been explored within metacreative systems: *top-down* approaches in which structure is initially generated, either by the system or the composer; and *bottom-up* approaches in which form emerges from the relationships between the details found within the music.

Perhaps most famously, Cope’s Experiments in Musical Intelligence uses a top-down approach to composition (Cope 2005). Brown, Wooller, and Miranda (2011) describe a system in which a composer can create material at landmarks within a composition and allow the computer to generate material between these points. Sorensen and Brown (2005) outline a system whose intent is to generate orchestral music in the aesthetic of mid- to late-Romantic eras. The system begins by generating a harmonic sequence to the length requested by the user (the authors suggest eight measures), then fills in other parts, such as rhythm, melody, and timbre. The authors acknowledge a lack of high-level structure in their system, which they claim to be a common weakness in generative music systems; however, the system does generate all aspects of music autonomously, beginning from the structural element of harmony.

Kuuskankare (2012) describes *Meta-Score*, a visual editor for PWGL, which defines structural, temporal, and procedural properties of a musical composition. *Meta-Score* is an off-line system that allows composers to place scored elements – which can be harmonic, rhythmic, or melodic material – on a time-line, and determine procedural dependencies between content. The author describes his system as Computer Assisted Composition (CAC), rather than as a generative system, as it allows the composer to continually refine the material; its ability to procedurally control aspects of structure are reasons for its inclusion here.

Maxwell created a cognitive model for music learning and generation (Maxwell et al. 2012), which, like *Meta-Score*, is a computer-aided composition tool. Of interest is its ability to learn higher-level relationships between musical phrases, and, in effect, generate formal properties. However, by their own admission, the system produces only limited structural representations.

Bottom-up approaches assume no knowledge of the future, and thus result in an emergent high-level structure. As noted, improvisational approaches are examples of dynamic formal generation, and the interaction between generative systems and composer/performers in real-time have a long history in computer music. Chadabe was the first to directly interact with musical automata: in 1971 he designed a complex analog system built by Robert Moog called the *CEMS*, which allowed him to compose and perform *Ideas of Movement at Bolton Landing*. This was the first instance of what he called *interactive composing*, “a mutually influential relationship between performer and instrument” (Chadabe 1984).

This notion of interaction remained important in such systems, so much so that the field became known as “interactive computer music” in the 1980s. Chadabe describes the relationship between composer/performer and system as such:

> the instrument is programmed to generate unpredictable information to which the performer reacts during performance. At the same time, the performer at least partially controls the instrument. Since the instrument influences the performer while, at the same time, the performer ‘influences’ the instrument, their relationship is mutually influential and, consequently, interactive (Chadabe 2007).

Composers such as Sal Martirano, David Behrman, Joel Chadabe, Martin Bartlett, Todd Winkler, Robert Rowe, George Lewis, and others designed software systems that made musical decisions in performance, under the influence of either a performer or operator (often the composer). Limitations to these approaches have been described elsewhere (Eigenfeldt 2007), specifically the limited role in which the computer can play in shaping the overall composition, which can be considered to be high-level musical decisions: in every case, these decisions remained under the control of the human. Drummond (2009) gives an excellent summation and classification of interactive systems.

Initial experiences in generating form

The author's research within metacreative musical systems began in the 1980s under the tutelage of Martin Bartlett, a pioneer of interactive systems, his own running on a Buchla 400. However, unlike Bartlett's systems, these rarely interacted with a performer; instead, they were attempts at real-time composition: a complex instrument capable of generating a multiplicity of gestures under the author's real-time control, with which a live performer could improvise (Eigenfeldt 1987, Eigenfeldt 2013). Regardless of who was driving the interaction – the performer in front of the microphone, or the composer holding the mouse – high-level *musical* decisions remained with the human.

As the level of complexity of the gestures– and the relationships and interactions *between* gestures – increased within these systems, a greater autonomy by the software in higher level musical decision-making was found to be necessary. While the use of multi-agents in *Kinetic Engine* created complex interactions that initiated on their own – thus reaching level 4 Proactivity on the metacreative taxonomy scale (Eigenfeldt et al. 2013) – the system still required a conductor to direct the musical progression in time (Eigenfeldt 2008).

Coming Together

The *Coming Together* series began to generate overall form, albeit in a limited fashion. *Coming Together: Freesound* (2010), *Coming Together: Beauty and Truth* (2010), and *Coming Together: Notomoton* (2011) involved musical agents altering their behavior over time, and thus moving to the next level of the metacreative taxonomy, level 5 Adaptability. *Freesound* and *Beauty and Truth* are both single movement compositions, which could be considered to involve pre-defined forms that are filled with novel material with each run: this reflects the use of templates by Colton et al. to produce poems (2012). *Beauty and Truth* is described in detail elsewhere (Eigenfeldt 2010), as is *Freesound* (Eigenfeldt and Pasquier 2011).

Notomoton is a multi-sectional work, in which the agent behaviors are determined by parametric limitations defined at the onset. Prior to beginning the work, a duration is set by the performer for the overall composition, and a five-part form of relative lengths is generated, in which a goal-level for density is determined, along with a tempo and tala (number of beats per phrase) for each section. Agents will attempt to achieve the set level for density – a complex measurement that involves the number of active agents, and the number of onsets played in the preceding two phrases.

This information is presented as a type of score to the improvising performer (see Figure 1); each performance inevitably begins with the generation of a form, and a quick judgment by myself regarding its aesthetic value, and re-generating forms until one appears that is considered useful: i.e. a high degree of contrast between sections, and a final section ending at a fast tempo and higher density.

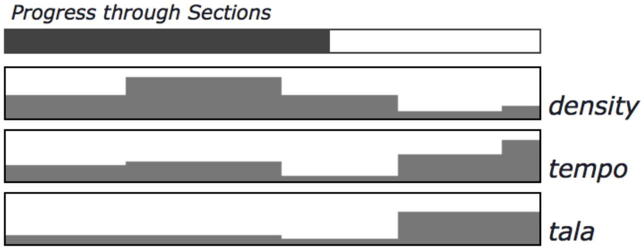

Figure 1. Generated form for *Coming Together: Notomoton.* In this score, the second section is the most dense, the third the slowest and with the shortest tala (phrase length), and the final section is the shortest in duration. Indications are relative, and based upon an overall duration (not shown).

While a clear form is suggested in the score, a great degree of variability results in performance, as the parametric score is one of *goals* for social agents. Furthermore, endings are the result of negotiation between agents: once the section passes 80% of its intended duration, the negotiation for an ending is initiated – a unison passage the outlines the tala; however, certain conditions must be present for all agents to participate in the negotiation, and sections can extend while beyond their intended length (or, end almost immediately after reaching the 80% point). An example performance of *Coming Together: Notomoton* can be viewed online (http://youtu.be/7HyU8nHs_pk), with percussionist Daniel Tones as performer.

A variation on this procedure is used in *More Than Four* (2012, described in Eigenfeldt and Pasquier 2012). In this work, the system uses the above described bottom-up methods to pre-generate a database of movements, ranging in length from one to four minutes. In concert, a curatorial agent then selects from these movements to create a unique multi-movement composition for that specific performance.

For the premiere performance, the curator agent used a somewhat simple scheme to select movements: choose a first movement that is bright and lively, then subsequently choose movements that are contrasting to immediately proceeding movements, ending with another bright, lively movement. This produced what the author considers to be a successful result – in this case, a series of movements whose relationships were interesting when performed consecutively; however, their selection, while unpredictable in their specifics, were predictable in the larger formal scheme.

Subsequent versions allowed for a freer selection process by the curator agent, based on a notion of similarity and dissimilarity of a variety of musical features between movements. This system is described more fully elsewhere (Eigenfeldt 2012); one pertinent observation should be mentioned here:

> What has proved interesting in performance has been how the eventual selections can evolve over time, due mainly to the roulette-wheel selection of rated movements. When a selection is made that is not at the top of the list, it may demonstrate certain aspects not found in those directly above it; as similarity (or dissimilarity) to this new object is made, aspects of the original criteria may no longer be present, and the system seems to wander towards its own preferences.

An example performance of *More Than Four* can be viewed online (http://youtu.be/ao_B0eW3gDI), with Daniel Tones and Brian Nesselroad: marimba; Martin Fisk, Timothy van Cleave, vibraphone; David Brown, bass.

These four works are examples of using a *bottom-up* approach to generate musical material while employing some *top-down* strategies to shape the structures. The author has found the use of social agents to be a powerful method of generating interesting and musically useful textural variation at a sectional level; in other words, in generating short forms of two to three minutes. However, the need to shape the overall composition, and to affect large-scale structural development requires some top-down methods; this desire is further complicated by the difficulties in constraining evolutionary methods within a typical concert work of ten to fifteen minutes.

Roboterstück

This work avoids the constraint of concert durations, and the necessity to control agent evolution, through its design as an ongoing installation. Not only does this allow for a greater variety of overall duration between different generations, but it allows gallery audiences to experience multiple versions of a generative work – something usually not possible in concert situations.

Roboterstück is performed by a mechanical musical instrument – the 18-armed *Karmatik NotomotoN* (Kapur et al. 2011) – and generates new compositions every 15 minutes, each of which last between three and six minutes. The composition is a tongue-in-cheek homage to Stockhausen's famous total-serialist work *Klavierstück XI*, in which the pianist glances at a sheet of music and randomly chooses to play from 15 notated fragments. In the case of *Roboterstück*, virtual agents negotiate a texture – from 16 possible combinations – based upon the following features: slow/fast; sparse/dense; loud/soft; rhythmic/arrhythmic. When the same texture has appeared three times, the performance is complete.

As with most of the author's multi-agent works, it is the social interaction between the agents that provides the immediate musical development; however, the overall form is condensed by agents having a boredom parameter. Once a texture has been negotiated and explored, agents slowly become bored; when more the half the agents have achieved this state, the section abruptly ends, thus providing a musically interesting and dramatic sectional change. Agents retaining a history of involvement further ensure variety between sections: if an agent participated in a previous section, it is less likely to join in on the next section.

An example performance of *Roboterstück* can be viewed online (http://vimeo.com/102746324), from an installation at NIME2014.

GESMI

Corpus-based methods in generating form are explored in *GESMI*, the Generative Electronica Statistical Modeling Instrument (Eigenfeldt and Pasquier 2013). This system generates complete dance music tracks based upon a corpus of 50 tracks of Breaks and House music, hand-transcribed by domain experts. Aspects of the transcription include individual instrumental parts and timbral descriptions, breaks and fills, and descriptions of overall musical form. This information was then compiled in a database, and analysed to produce data for generative purposes.

Form within the selected EDM genres is quite predictable when compared with other musical genres. The styles analysed use 8-bar phrases to a very high degree of consistency: for example, less than 2% of the 621 phrases in the Breaks corpus are something other than 8 bars in length. Furthermore, the entire corpus can be delineated into five distinct formal sections, which we label A-E:

A- Lead-in: the initial section with often only a few parts present;

B- Intro: a bridge between the Lead-in and the Verse. More instruments are present than the Lead-in, but not as full as the Verse;

C- Verse: the main section of the track, in which all instruments are present, which can occur several times;

D- Breakdown: a contrasting section to the verse in which the beat may drop out, or a filter may remove all mid– and high–frequencies. It will tend to build tension, and lead back to the verse;

E- Outro: the fade-out of the track.

While each corpus track can thus be described by the order of these units – i.e. AABBBBDCCC – the complexity of the music is clearly not captured: its subtle formal variation is found in how the individual patterns are introduced and varied within this larger structure. For this reason, a genetic algorithm was used to generate not only the overall form, but the relationships between the parts within the macro-structure. Once the number of patterns per part, and their specific locations and relationships are established, the actual patterns are generated for each instrument, knowing with which patterns they must interact.

As such, *GESMI* is an example of a successful top-down approach to generative composition. What was found to be most satisfying in the generated music was the clear sectional repetitions and formal outlines that could be heard, while maintaining variation at the phrase level; of course, this was to be expected through the use of a top-down approach, but it was nevertheless unusual within the author's generative music. The system is more fully described elsewhere (Eigenfeldt and Pasquier 2013). Example tracks from GESMI can be heard online (http://soundcloud.com/loadbang).

The Indifference Engine

The author's most ambitious exploration of autonomous formal design occurs within *The Indifference Engine* (2013). This system attempts to create true BDI (belief-desire-intention) agents (Rao and Georgeff 1995). Using a multi-agent system that involves negotiation amongst the virtual agents, this system can include a live performer as one of the agents.

Desires & Intentions

The basic concept behind *The Indifference Engine* is several independent agents - each of which has independent goals - attempting to negotiate a successful musical improvisation, which includes a negotiated overall structure. Just as human improvisers must agree beforehand on an (approximate) performance length, a overall duration is initially set (by the user, but this could easily be a negotiated feature). Given this timeframe, agents derive independent intentions for pitch, amplitude, and speed (see figure 2).

These three separate intentions are then averaged to form the agent's overall *tensionCurve* (see Figure 3), also considered the agent's "intention". Two additional parameters - rhythmicity and density - are derived through real-time Brownian motion. Note that curves do not necessarily begin low, move high, and then end low again.

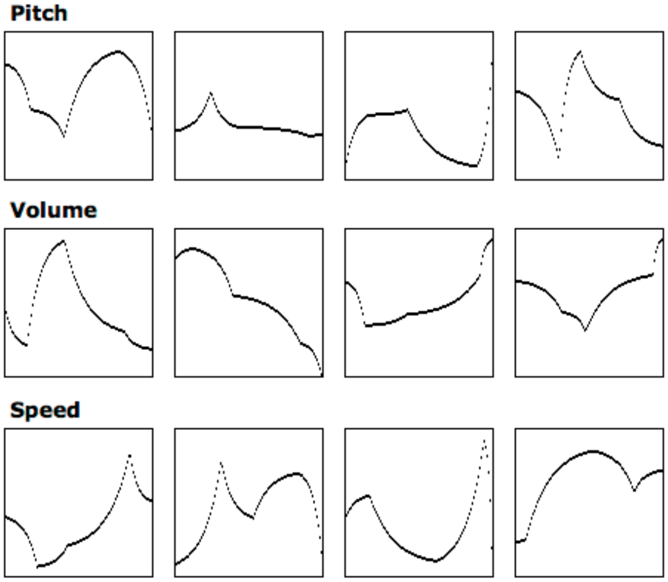

Figure 2. Pitch, Volume, and Speed contours for four individual agents, generated at initialization.

A single agent, without any social interaction, would generate its musical parameters based upon its intentions. Within a social environment, agents share their generated data (rather than their intentions, which remain hidden). A mean value for each parameter is calculated every *checkTime* (described below) and globally stored; agents compare the ensemble values to their own intentions. Agents closer to the mean receive a higher *confidence* score while more outlying agents receive lower *confidence* scores. These scores are used by agents to decide whether to continue with their own intentions, or adjust their intentions closer to that of the ensemble.

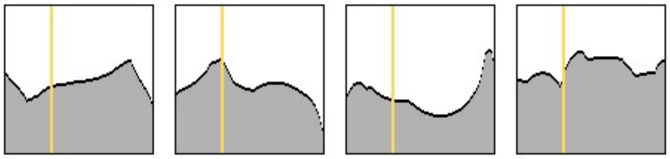

Figure 3. Four *tensionCurves*, for four different agents, based upon averaging out the agent's pitch, amplitude, and speed contours. Time is on the *x*-axis: amount (0.0 - 1.0) is on the *y*-axis. The yellow vertical bar shows the progress through the composition.

Proactivity

When the performance is initiated, agents generate a random interval to test their environment, an exponential random value between 50 and 4000 ms that is their *checkTime*. Agents only change their behaviors during this check; the independence of check times ensures agent actions occur independently. De-syncing agent decision-making was first explored within *Kinetic Engine* (Eigenfeldt 2008).

Agents become active by testing their current intention against the percentile of active agents. Thus, once the performance begins, the first agents to test their environment will be the first to become active. In order to avoid agents spuriously turning on and off, four positive tests in a row must occur to turn on, while four negative tests will turn the agent off.

When an agent becomes active, it notes its current mode: *internal* or *external* (explained shortly), and rewards that mode. Agents that become active while looking internally will begin to favor that mode; agents that become active while listening to the live performer will tend to favor that mode.

Negotiation

Additional negotiation occurs between the agents on every *checkTime* to determine the agent's *playingStyle*, which is the current binary state for pitch (low/high), amplitude (soft/loud), rhythmicity (rhythmic/arrhythmic), density (spare/dense), and speed (slow/fast). For the purposes of negotiation, agent's current intentions for these parameters – floating point values between 0.0 and 1.0 – are rounded to binary states of 0 or 1.

Negotiation is accomplished by randomly selecting an active partner, and comparing intentions (rounded to binary states). One parameter from the non-matching styles is then accepted (see Table 1). While this results in agents playing in a style that may not reflect their intentions, the next negotiation will begin again from the agent's intention values, rather than the previous playing style.

	pitch	**amp.**	**rhythm**	**density**	**speed**
agent 1 (intention)	0.35	0.55	0.25	0.85	0.75
agent 1 (playing style)	0	1	0	1	1
agent *n*	1	0	0	0	1
new agent 1	0	0	0	1	1

Table 1. Negotiating playing style for agent 1. Row 1 shows intention values for an instant in the composition; row 2 shows these values translated to binary playing style values; row 3 shows playing style values for another agent; row 4 shows new playing style values for agent 1, with a new negotiated value for amplitude. The agent will play "soft" until the next *checkTime*, despite its amplitude intention being closer to "loud".

This negotiation is further complicated by an agent's *persistence.* During the negotiation process, this value is tested against a randomly generated value; if the random value is not below the agent's *persistence*, the agent will forgo a round of negotiation. Once negotiation is complete, and all agents have the same playing style, that negotiated style is compared to the original playing style of the agent (derived from their intentions): those agents with four out of five matches have their persistence reinforced; those with less have it lowered.

Additional negotiation occurs for the *gesturalEnvelope,* and the *currentCorpus*.

Synthesis

A modified version of *CataRT* (Schwarz 2007) is used for sound generation, which requires a corpus of samples; parameters for a granular synthesis engine at both a micro- and macro- level are generated. Within *CataRT*, a number of different samples can be analysed, which results in the samples being broken down into individual grains (the default of 242 ms grain duration is used), and distributed on a plot based upon pitch (x) and loudness (y).

In performance, up to eleven different corpora have been used, each of which has been further analysed for various features, including volume (soft/loud), frequency (low/high), density (sparse/active), and rhythm (arrhythmic/rhythmic) as well as a 24-band Bark analysis (Zwicker and Terhardt 1980). Agents are initially assigned a random corpus; during negotiation, agents compare their corpus to other agents (using a Euclidean Distance Function on the above analysis data), and attempt to converge on a single corpus. Once this occurs, agents will select a new corpus, and begin again.

When an agent is active, it selects the actual grain samples using *CataRT* based upon its pitch (x) and loudness (y) intention; however, this is influenced by the agent's *confidence*, which is determined by its relationship to other agents (described earlier). The agent that has the highest confidence – e.g. closest to the group mean – becomes the leader, and that agent's pitch/loudness point is used as a point of attraction for a *Boids* algorithm (Reynolds 1987) to move through the Cartesian plane. Agents with a high confidence score will tend to retain their own location; agents with a lower score will follow an assigned flocking boid.

Internal vs. External: Arguing vs. Listening

The live performer is treated as a special agent: the performer's audio is analysed for the same features as the corpora: volume, frequency, density, rhythm, and a 24-band Bark analysis. Additionally, the system is trained on at least three playing styles of the performer – plus silence – using a simple neural network. The previous performances have involved a percussionist/drummer, and the learned playing styles included skins (drums), metal (cymbals), wood (rims), plus silence. The system selects the best match using a Distance Function from the Zsa descriptors (Malt and Jourdan, 2008), and provides a confidence rating in this timbre.

The analysis data is stored in a rotating buffer of two seconds duration, to which each agent can only access a unique 250 ms slot. In other words, each agent must form a belief of the live performer's current state based upon a limited window. Clearly, this window can quickly change and provide contradictory information to the different agents; as a result of this distorted viewpoint, the agents spend a lot of time arguing.

Lastly, each agent must decide whether to follow the group dynamic (an *internal* view), or the live performer (an *external* view). As each agent's immediate goal is to be active, the state the agent is in (*internal* vs. *external*) when the agent becomes active is rewarded. In certain situations, a majority of the agents will have an internal state, and thus appear indifferent to the performer.

The Indifference Engine: Summary

As with most of the author's metacreative systems, a great deal of time is spent fine-tuning the parameters in an effort to find those "sweet-spots" that are rewarding musically, yet surprising. *The Indifference Engine* has proven to be a very dynamic system that behaves in extremely complex ways: it tends to "go off on its own", yet retains references to the live performer, following performance gestures – particularly timbral changes – in fascinating ways. Performers have found it interesting to work with the system, despite the fact that it does often appear rather indifferent.

As the agents end up following a negotiated formal structure that revolves around their generated intentions, a formal shape does emerge; however, I'm hesitant to consider it truly autonomous: certain types of structures will emerge due to the heuristics of the implementation, while others simply will never appear.

The premiere of *The Indifference Engine* can be viewed online (http://youtu.be/o0Q_wEi0AF0), with percussionist Brian Nesselroad as performer.

Conclusion and Future Work

This paper has described some pragmatic approaches to dealing with generating large-scale musical structure. These include both top-down approaches – which allow for the specification of relationships of parameters within a section as well as between constituent elements of the overall composition – and bottom-up approaches, which allow for the dynamic development of the materials themselves. The BDI model described in *The Indifference Engine* is a unique exploration and contribution, which balances an overall negotiated form with dynamic relationships that evolve within the performance in response to a changing environment. However, its latent structures are constrained by the structure of the program itself.

The delineation of time within music remains a complex task for metacreative systems. While top-down solutions remain viable for certain contemporary genres – dance music, for example – a balance between such predictable structures and the dynamism of bottom-up approaches will be necessary for new applications of generative systems within interactive music contexts, such as computer games and streaming media services. The non-linearity of these forms already explore the potential for generative techniques (Togelius et al. 2007), including, to a limited degree, that of music (Collins 2009). The promise of a metacreative system that can autonomously generate complex music that demonstrates interesting large-scale structure, coupled with the ability to adapt to a dynamic environment, remains an open problem.

References

Boulez, P., Noakes, D., Jacobs, P. 1964. Alea. *Perspectives of New Music* 3:1, 42–53.

Brown, A., Wooller, R., Miranda, E. 2011. Evolutionary morphing for music composition. *A-Life for Music: Music and Computer Models of Living Systems*, Vol. 24. AR Editions, Inc., 115–132.

Chadabe, J. 2007. "Electronic Music: Unsung Revolutions of the 20th Century" http://www.percontra.net/6music2.htm. Accessed June 13, 2007.

Chadabe, J. 1984. Interactive Composing. *Computer Music Journal* 8:1.

Collins K. 2009. An Introduction to Procedural Music in Video Games. *Contemporary Music Review* 28(1): 5–15.

Colton, S., Goodwin, J., and Veale, T. 2012. Full face poetry generation. In Proceedings of the Third International Conference on Computational Creativity, 95–102.

Cope, D. 2005. *Computer Models of Musical Creativity*. Cambridge: MIT Press.

Drummond, J. 2009. Understanding Interactive Systems. *Organised Sound* 14:2, 124–133.

Eigenfeldt, A. 1989. ConTour: A Real-Time MIDI System Based on Gestural Input. International Conference of Computer Music, Columbus.

Eigenfeldt, A. 2007. "Realtime Composition or Computer Improvisation - A Composer's Search for Intelligent Tools in Interactive Computer Music", EMS07 Leicester, http://www.ems-network.org/spip.php?article264, accessed 6 December 2009.

Eigenfeldt, A. 2008. Emergent rhythms through multi-agency in Max/MSP. *Computer Music Modeling and Retrieval. Sense of Sounds*. Springer Berlin.

Eigenfeldt, A. 2010. Coming Together - Negotiated Content by Multi-agents. Proceedings of the ACM Multimedia International Conference, Firenze, 1433–1436.

Eigenfeldt, A. 2011. *Real-time Composition as Performance Ecosystem.* Organised Sound 16:2, 145–153.

Eigenfeldt, A,. and Pasquier, P. 2011. Negotiated Content: Generative Soundscape Composition by Autonomous Musical Agents in Coming Together: Freesound. In Proceedings of the Second International Conference on Computational Creativity.

Eigenfeldt, A. 2012. Embracing the Bias of the Machine: Exploring Non-Human Fitness Functions. In Proceedings of the Artificial Intelligence and Interactive Digital Entertainment (AIIDE'12) Conference, Palo Alto.

Eigenfeldt, A., and Pasquier, P. 2012. Creative Agents, Curatorial Agents, and Human-Agent Interaction in Coming Together. In Proceedings of Sound and Music Computing, Copenhagen.

Eigenfeldt, A., and Pasquier, P. 2013. Evolving Structures in Electronic Dance Music. In Genetic and Evolutionary Computation Conference (GECCO), Amsterdam.

Eigenfeldt, A., Bown, O., Pasquier, P., Martin, A. 2013. Towards a Taxonomy of Musical Metacreation: Reflections on the First Musical Metacreation Weekend. In Proceedings of the Artificial Intelligence and Interactive Digital Entertainment (AIIDE'13) Conference, Boston.

Galanter, P. 2003. What is Generative Art? Complexity theory as a context for art theory. In GA2003 6th Generative Art Conference.

Hindemith P. 1970. *The Craft of Musical Composition: Theoretical Part 1*. Mainz, Germany, Schott.

Huron, D. 2006. *Sweet anticipation. Music and the Psychology of Expectation*. Cambridge, MA: MIT Press.

Kapur, A., Darling, M., Murphy, J., Hochenbaum, J., Diakopoulos, D., Trimpin 2011. The KarmetiK NotomotoN: A New Breed of Musical Robot for Teaching and Performance. In Proceedings of the 2011 Conference on New Instruments and Musical Expression, Oslo, 228–231.

Kuuskankare, M. 20012. Meta-Score, a Novel PWGL Editor Designed for the Structural, Temporal, and Procedural Description of a Musical Composition. In Proceedings of the Artificial Intelligence and Interactive Digital Entertainment (AIIDE'12) Conference, Palo Alto.

Lerdahl, F., Jackendoff, R. 1985. *A Generative Theory of Tonal Music*. MIT press.

Malt, M., Jourdan, E. 2008. Zsa.Descriptors: a library for real-time descriptors analysis. In Sound and Music Computing, Berlin, Germany.

Maxwell, J., Eigenfeldt, A., Pasquier, P., Thomas, N. 2012. MusiCOG: A cognitive architecture for music learning and generation. In Proceedings of the Sound and Music Computing Conference.

Meyer, L. 1956. *Emotion and meaning in music.* Chicago: University of Chicago Press.

Narmour, E. 1999. Hierarchical expectation and musical style. *The psychology of music.* New York: Academic Press, 441–472.

Negretto, E. 2012. Expectation and anticipation as key elements for the constitution of meaning in music. *Teorema: Revista internacional de filosofía*, 31(3), 149-163.

Rao, A. and Georgeff, M. 1995. BDI-agents: From Theory to Practice. In Proceedings of the First International Conference on Multiagent Systems (ICMAS'95), San Francisco.

Reynolds, C. 1987. Flocks, herds and schools: A distributed behavioral model. *ACM SIGGRAPH Computer Graphics,* 21(4), 25–34.

Schenker, H., Jonas, O. 1972. *Harmony*. Edited and annotated by Oswald Jonas. Translated by Elisabeth Mann Borgese. University of Chicago Press.

Schwarz, D. 2007. Corpus-based concatenative synthesis. *IEEE Signal Processing Magazine*. 24(2), 92–104.

Sorensen, A. and Brown, A. R. 2008. A Computational Model For The Generation Of Orchestral Music in the Germanic Symphonic Tradition: A progress report. In Proceedings of Sound : Space - The Australasian Computer Music Conference, Sydney. ACMA,78–84.

Temperley, D. 2001. *The cognition of basic musical structures.* Cambridge, MA: MIT Press.

Togelius J, de Nardi R, Lucas, S. 2007. Towards automatic personalised content creation for racing games. In Computational Intelligence and Games, IEEE.

Togelius, J., De Nardi, R., Lucas, S. 2007. Towards automatic personalised content creation for racing games. In IEEE Symposium on Computational Intelligence and Games.

Zwicker, E., Terhardt, E. 1980. Analytical expressions for critical-band rate and critical bandwidth as a function of frequency. *Journal of the Acoustical Society of America*. 68(5), 1523–1525.