Andrew Sorensen
Queensland University of Technology &
Australasian CRC for Interaction Design
Kelvin Grove, 4059, Australia
a.sorensen@qut.edu.au

Andrew R. Brown
Queensland University of Technology &
Australasian CRC for Interaction Design
Kelvin Grove, 4059, Australia
a.brown@qut.edu.au

A Computational Model For The Generation Of Orchestral Music In The Germanic Symphonic Tradition: A progress report

Abstract

In this article we report on progress at the Australian CRC for Interaction Design investigating the computational generation of orchestral music based in the Germanic Symphonic tradition. We present an introduction to the project including a brief overview of our intended methods and some guiding principles for the project. We then outline the current state of the project and introduce our initial algorithmic system with a special emphasis on an implementation of Paul Hindemith's harmonic system. We conclude with some initial findings and future goals. We provide an extensive selection of audio examples online that accompany, verify and enhance information provided in this report.

Introduction

Starting in 2008 the Australian CRC for Interaction Design began an exploration into the computational generation of orchestral music based loosely around a mid to late romantic aesthetic. There are several reasons for our current interest in computational approaches to the generation of romantic orchestral music. Firstly, there is a commercial aspect to this investigation. A significant amount of the music currently used in mainstream media is orchestral; feature films, documentaries, television dramas/sitcoms and computer games all make significant use of orchestral soundtracks. Yet the computer plays a disproportionally small role in the creation of orchestral music beyond it's role as a notation and recording device. Arguably this is due to the structural complexity of orchestral music of the mid-late romantic period. This leads to a second interest in the project, an intellectual interest in what continues to be a contentious area of investigation. While we acknowledge the significant efforts of projects such as Cope's EMI software (Cope 1992, 2001, 2005), there remain many problems with the application of such research within a real-world context. Cope's work is non-realtime, requires a substantial database of painstakingly analyzed works and provides limited performance playback. A cursory survey of current professional music software reveals the limited practical real-wold impact of current algorithmic composition systems and research. Thirdly, we have an aesthetic interest in music of this genre and although our own musical practice extends well beyond this one genre, there is a fundamental musical vocabulary here that we perceive as integral to our development as computational musicians.

Background

After many years of building computational music systems the authors are now guided by two beliefs.

Firstly, the seemingly obvious belief that algorithms derived from musical analysis are far more likely to provide effective musical solutions than those from formalisms outside the music domain. After spending many years investigating Neural Nets, Genetic Algorithms, Genetic Programming, and the like (Towsey et al. 2000; Sorensen 1998; Brown 2005, 2005a; Brown et al. 2008) the authors have returned to a simple set of principles based around probability, linearity, periodicity, set theory and recursion (Sorensen and Brown 2007). This is not to say that there is not or will not be a role for the aforementioned processes, only that these algorithms will be used for targeted solutions when directly musically applicable. Our goals are first and foremost achieving a musical outcome and although we may stumble across issues related to machine or human creativity these are not primary goals of the project.

Secondly, we are interested in holistic solutions. We question the usefulness of a reductionist approach that focuses on one musical element, e.g., pitch, rhythm, melody, harmony or timbre, in isolation when dealing with real world compositional problems. We anticipate that we will be better able to address sub problems while experimenting with the whole.

Method

At this stage of the project we have been working loosely along the lines of the scientific method whereby various musical hypothesis, conceptualized as program code, are tested for observable, empirical results. Indeed the datum of our project, are the musical compositions generated by our system.

It is, however, the authors belief that objective measurement, one of the cornerstones of the scientific method, is not only impossible in assessing music composition given the aesthetic nature of our data , but is also dangerous as it can lead to rational explanations that do not bear experiential scrutiny (Brown 2004). Measurement must therefore take place within an aesthetic framework, and rely upon the subjective judgment of the researcher who will make his results available for critique and comment. In this spirit we provide ample examples of data from our experiments online for peer review in an attempt to temper the necessarily subjective nature of our experiments. We hope that this will go some way to provide the reader with confidence in our methods and results.

Hindemith's Harmonic System

An initial starting position for the project was to search for an harmonic system suited to the greater chromatic content of the germanic symphonic musics of the later half of the nineteenth century. Not only would the chosen harmonic system need to be musically applicable but also implementable within the constraints of a realtime system.

Paul Hindemith's "The Craft of Musical Composition - Book 1" outlines an harmonic system based around intervalic relationships and the tensions inherent in these relationships. While certainly not the only comprehensive theory based around intervalic relationships Hindemith's is simple, elegant and suitable for fairly direct computational implementation. Importantly for our chosen genre Hindemith's system is a tonal, but non-diatonic system, giving us a good balance between harmonic freedom and control.

It is worth noting that Hindemith's tension relationships are subjective and some have argued about the theoretical soundness of his choices because of a reliance on harmonic relationships that may only formally be applicable to music using just intonation or that in an equally tempered system blur functional distinctions between major and minor triads (Clark 2007). Nevertheless his system presented a good starting point because it afforded rapid implementation and subsequent aural experiment.

Our initial investigation of Hindemith's system indicated that it would provide musical benefits including; chromatic flexibility, relatively direct control of musical tension, the ability to cleanly separate chord quality from root progression and the trivial variation of existing root progressions. Some of the technical advantages of the system are it's suitability for realtime work, it works within limited temporal restraints (i.e. no need to look ahead or behind) and the general efficiency of the algorithm. It is likely that we will add to this harmonic system in the future, particularly once we have broadened the stylistic range of our experiment.

Another strong validation of Hindemith's harmonic system is Hindemith himself who made extensive use of the system for his own compositions, going so far as to rewrite many of his early works so that they would conform to this system which he developed later in life. This suggested not only a practical solution, such that he was able to retrofit existing works, but also a musically meaningful solution given that he would bother to make the effort and that works made with this system are held in high regard.

Our intentions were never to test Paul Hindemith's system as originally conceived but to use it as a starting point to further our own harmonic enquiry. Hindemith never intended for the system to be autonomous and as such it provides a general, but not absolute, guide for us to follow. We are therefore adding, modifying and removing aspects of Hindemith's system as our test results dictate.

Hindemith's System Of Chord Qualities

Hindemith devised a system of chord qualities that grouped all possible chords into one of six chord-groups, numbered I-VI. Hindemith's system segregated chords based upon the intervalic relationships between chord tones. The combined set of internal intervals identify the chord-group to which the chord is assigned. Chords in a particular group are assumed to share a similar pureness, stability or harmonic tension. The six groups are defined as follows:

I] No tritone, no seconds, no sevenths
II] No minor seconds, no major 7ths, with tritone
III] No tritones
IV] Contains one or more tritones
V] Indeterminate
VI] Indeterminate (with tritone predominating)

As an example consider the dominant 7th chord C7 [C E G Bb] which contains the following set of intervals [M3 P5 M6 m6 m7 m3 M2 tritone]. This interval set places C7 in chord-group II. Cmaj7 [C E G B] which contains the intervals [M3 M6 m6 P5 m7 m3 m2] would be placed in chord-group III.

Hindemith makes additional sub-segregations based on root position although we ignore this addition in our present work. Hindemith also provided a guide for the creation of root progressions by again appealing to intervalic relationships. Hindemith defined two interval series (series I and II) that he uses for the creation of root progressions and other melodic materials. We do not provide further explanation here as our present system makes only indirect use of these series for the creation of root progressions.

In a simple sense Hindemith's theory progresses by assigning chord quality to root notes where the choice of chord quality is made based upon the amount of harmonic tension required at any given point in the composition. Changes in tension or resolution are relative to the current chord quality.

Modifications to the Harmonic Model

During the development cycle there have been a number of technical additions made to provide constraints over Hindemith's system. Firstly there is a Pitch Class Set (PCS) parameter that offers the opportunity to help constrain chordal choice. This provides a level of diatonic support not explicitly available in Hindemith's system. For example, Hindemith's system places no constraints over the choice of major or minor chord qualities, leaving this instead as a subjective choice for the composer. For example, both Cmaj [C E G] and Cmin [C Eb G] are contained within chord-group I. Our implementation uses a PCS parameter as a means for weighting these types of choices. In this sense the PCS imposes a key over the music although we provide the ability to weight this imposition (i.e. we don't always apply the PCS).

Additionally, the current implementation of the algorithm only makes use of Hindemith's first three chord groups (I-III) currently leaving out the indeterminate choices (V-VI) and the multiple tritones subordinate group (IV). This was an early choice to impose limits on the range of harmonic variety. To date, our limited subset has proven to provide adequate variation. We may choose to add the further 3 groups at a later stage or for use in different musical contexts. We also do not currently implement inversions as distinct chordal subgroups, as Hindemith does. This allows us to more simply manage the construction of musically meaningful bass movement.

System Overview

Before beginning a detailed investigation of the algorithmic implementation we will describe in brief the overall design of music production. The entry of our process begins with harmonic selection. There is no requirement to begin the compositional process with harmonic selection, melodic or rhythmic materials provide equally valid starting points. However, given the dominance of harmonic consideration in late romantic music and the emphasis on thematic rather than more traditionally melodic passages (Crocker 1966), starting with harmonic structure appeared a reasonable decision.

Our harmonic selection process begins by generating a root progression including both pitch and rhythm dimensions. The root progression is generated to fit a given time period, eight bars for example. The algorithm then progresses linearly through the root progression. For each root note several activities take place. Firstly a chord quality is selected for the given root. The harmonic tension of this chord quality is substantially influenced by the Hindemith chord group value (I, II, or III) which is chosen based on the current harmonic tension required. Once the chord voicing is generated a score is constructed based on current settings for orchestration, pitch range, harmonic accompaniment and constrained by voice leading requirements. Finally, a bass line is added and thematic material is selected and manipulated to fit the current harmonic context.

Of the many aspects of this process the most highly developed relate to the earlier harmonic considerations with little work started on most of the later stages. However, as previously mentioned, one of the primary hypothesis of this project is that by developing all aspects of the system concurrently we will be able to move forward more successfully across the whole project. We have therefore made an effort to include algorithms covering all aspects of the process as early as possible. One interesting result of this experience is just how much is possible with a very superficial implementation provided that coverage is broad. We will now discuss each aspect of the overall system in more detail.

Rhythm Generator

The rhythm generator is a simple stochastic function providing control over duration, tactus, level of syncopation and rhythmic value list (herein called the RVL parameter). The generator initially selects a rhythm value at random from a supplied RVL. The generator will continue to select rhythm values at random from the RVL until either (a) the maximum duration is reached or (b) the maximum duration is surpassed at which point the algorithm will backtrack to a point at which it can continue forward.

The musicality of this algorithm is provided by incorporating the gestalt laws of proximity and similarity. A gestalt selection in the rhythm generator forces a re-selection of the previous rhythm value, providing local coherence. The weighting of this gestalt selection is made through a combination of the tactus and syncopation parameters. At a tactus point a stochastic selection will always be made from the RVL. At all other times the syncopation value (a percentage value), will determine whether a gestalt selection of the previous rhythm value or stochastic choice from the RVL. Syncopation in the algorithm is largely influenced by the selection of rhythmic values that cross tactus points.

This simple algorithm has proven very effective at handling all of our existing rhythmic generation requirements.

Root Selection

The root selection algorithm currently operates using a relatively simple point-to-point style approach. There is no explicit cadential knowledge built into the root selection algorithm. The first part of the root selection process defines a harmonic rhythm selected using the stochastic rhythm generator described above. The point-to-point process begins with parameters for starting Pitch Class (PC), ending PC and an Interval Value List (IVL). The algorithm selects N-2 (i.e., N being the number of rhythmic values generated by the rhythm generator, minus starting PC and ending PC) values from the IVL that when combined move from starting PC to ending PC.

The IVL is critical to the harmonic complexity of the resulting root progression. For example, an IVL containing only 4ths and 5ths will result in very conventional root movement. Adding additional intervals to the IVL will significantly alter the harmonic complexity of the root movement. This highlights the importance of the root progression and it's equally important role to the chord quality selection process for managing harmonic complexity.

To clarify the root progression process let's assume a simple rhythm of (4 4 4 4 4 4) where each number represents beats per root. If we proceed with a starting PC or 0 (C), an ending PC of 7 (G) and an IVL of fifths and fourths (7 5) our root generator might (this is a probabilistic process) return the list (0 5 7 5 7 7). Once these intervals are summed and translated we get the root-progression (C F C F C G). We might also have received (0 7 7 7 5 5) which would give us (C G D A D G). The important things to note are (a) we are working with PC's so everything is mod12 (b) that the generated progression always moves from starting PC to ending PC (c) that changes to the IVL will significantly alter the complexity of the root progression.

Cadential knowledge can be added to this process by setting an appropriate PC end point and appending a resolving PC. A half cadence for example requires no modification to the existing process, requiring only that the first PC is the root and the final PC is the fifth. A perfect cadence is implemented by appending a resolving I chord. This simple process provides the basis for implementing all major cadential forms. The critical roll that bass movement plays in cadential movement is acknowledged and is discussed later.

As discussed above, the harmonic complexity of the root progression is controlled by the contents of the IVL, however two further Pitch Class Set (PCS) quantization methods are available. Firstly, the output of the root progression generation can be quantized against a PCS. Secondly the IVL can be implemented as a Step Value List (SVL) rather than an Interval Value List (IVL). This automatically places all results within the current key as the SVL is bound to the key's church mode. Both of these techniques provide us with the ability to vary constraints control over the harmonic output.

In conclusion, our control over the root selection process is fairly fine grained. We can choose to quantize to a PCS or not, we can choose to work within a traditional system of cadences or not, we can work with IVLs or SVLs and we can control the degree of harmonic complexity in our root progression by adjusting the values in the IVL/SVL.

Chord Quality

A chord quality is defined for each root in the root progression in realtime. The chord quality is selected using Hindemith's chord quality system. The chord quality selection process proceeds as follows.

A random selection of pitch classes is made and the complete set of intervals joining these pitch classes is calculated. For example the selection [C E G] would produce the interval list [M3 m3 P5 M6 m6]. This set of intervals is then tested against whichever Hindemith chord group is currently required to fulfill the harmonic tension requirement - moving from least tense I to most tense III. Successful validation of the random selection moves us to the next stage where we assign the required root to the chord. Remembering that Hindemith chord groups are interval based, that is they have no specific pitch requirements, we can freely transpose the returned chord to whatever our root requirement is. Using the example above, [C E G] will pass validation against chord-group I. If this was indeed the chord-group we required then [C E G] will be returned to us and we can then transpose this freely to a required root. A root of D for example would result in [D F# A].

At this stage the PCS does not designate an inversion nor octave displacement (i.e. it is an unordered set of pitch classes). It is also very important to note that chord-group I would be equally as likely to successful return [D F A]. This is where we can choose to apply key constraints, by quantizing the result to a PCS church mode matched to the current key. Quantizing [D F A] against the PCS [2 4 6 7 9 11 1] will force [D F A] to [D F# A].

Bass Note

After assigning a chord quality the next stage of the process assigns one or more bass tones for the current chord duration (the bass may move). For cadential purposes the generator attempts to accentuate the current musical key while avoiding strong cadential reference. Our initial implementation achieves this by using a simple weighting to reject root and bass correlation (i.e., we use inversions instead of root positioning) away from phrase boundaries. At this stage we freely invert using any option available within the context of the current chord quality. The generator also uses tonic inversions where possible.

Scoring

The next phase involves the voicing and orchestration of the chord. We now have a fairly complete harmonic picture, we know the root, quality and bass (i.e. inversion) of the chord. Using this information we proceed to orchestrate the chord using a further range of system parameters including the current instrumentation, number of active voices, and lower and upper pitch bounds. Additionally the scoring algorithm uses the previously scored harmonic block as a reference point for smooth linear part movement.

Our linear part movement algorithm follows a shortest path approach searching for the shortest path between two given chords while maintaining complete chord coverage (i.e., making sure each chord tone is represented). Finally, the algorithm makes decisions about the style of accompaniment to apply and what instruments may double parts etc.. At present we have only implemented two accompaniment styles, an arpeggio style and a homophonic style. We plan to greatly expand the number of accompaniment styles in the future.

A choice about whether or not to perform the bass part is also now made depending upon the instrumentation, range and accompaniment style chosen.

Thematic Material

The last stage in the generation of note data involves the generation of a thematic fragment to fit the current harmonic context and duration. A theme is created by generating a rhythm, using the stochastic rhythm generator, of equal duration to the current chord, and selecting intervals at random from an IVL. The generated theme can then be transposed to commence on any degree of the currently active chord. The theme is quantized to a suitable scale - a suitable scale being a scale that agrees with the current chord quality, chord root and current key. As well as generating new themes the system also has the ability to reuse existing themes providing a level of cohesion both locally, through simple cannoning, and more globally by reintroducing previous themes.

The scale selected for theme quantization is chosen by analyzing the chord and comparing this against the standard church modes rooted against the current key. For example, the scale selected for G7 [G B D F] in the key of C major would be G Mixolydian.

There is an additional option to root the modes against the chord root which is often applied against chordal roots outside of the current key. This helps to emphasize the non diatonic influence of the chord.

With a series of simple extensions this trivial thematic generator can be tuned to provide passable material. In part we feel this is due to the dominance of thematic rather than melodic invention in romantic music (Crocker 1966).

We provide the ability to choose not to play a melodic fragment and provide the option to accompany the chosen thematic fragment with a delayed copy played on a second instrument, possibly in a separate register. The final stage of thematic generation is to assign an instrument for the performance of the theme.

Orchestration

After the thematic material has been generated our musical data is ready to be performed. Impromptu supports the AudioUnit specification and we use the Vienna Symphonic Library for sample playback. Being a fully functional AudioUnit host, Impromptu provides us with complete control over this playback including the ability to freely modulate performance parameters at runtime.

We have designed an quasi agent-based approach for instrumental playback to take advantage of this low level control. Instruments act semi autonomously in their responses to musical note information choosing which sample patch to use at what time based on note volumes, duration, articulation styles, and so on. For example, a trumpet knows which sample bank it should play from given a heavy attack, loud volume and short duration, as well as providing some simple range checks and instrument specific options (muted options for brass, pizzicato for strings etc..). We also apply a variety of gestural control mechanisms at this level, such as multi layered oscillators for dynamic modulation, control of legato performance parameters, fine grained volume control and alike as detailed previously (Sorensen and Brown 2007). These performance technique greatly enliven performance.

Additionally the playback system supports multiple independent metronomes allowing us to modulate tempo for each individual part if required. This allows us to fairly trivially add rubato playing where instruments are linked to a single metronome or move independently to their own individual metronome.

The process described above is rapid enough to be calculated and performed in real-time for a relatively complete orchestration of flutes, oboes, clarinets, bassoons, trumpets, horns, trombones, tuba, violins, violas, cellos, basses and percussion.

To discuss in detail all aspects of our performance system would require another paper of equal length but we would like to acknowledge the important contributions of Clynes (1984) and Desain and Honing (1993) to this work. We intend to present our performance system in greater detail at sometime in the future.

Shortcomings

Before we begin a discussion about the results of our current research we would like to point out some of the musical shortcomings inherent in our current system such that we can pay particular attention to them when discussing our results.

We make very little allowance for good voice leading beyond our simple shortest path solution. Voice leading is an area in which computational study has arguably had it's greatest successes (Ponsford 1999, Huron 2001, Hömel 2004). Given that we pay only superficial regard to voice leading we can expect to suffer from many of the classic part writing concerns, parallel movement, consecutive fifths and octaves etc.. Another concern of our current shortest path algorithm is our lack of control over the distribution of voices over the complete pitch range. In other words there is little to stop bunching of parts at the top, centre or bottom of the pitch range.

There is no explicit shaping of the point-to-point root progression selections which could result in non-directed root progressions and therefore non-directed harmonic movement. Additionally, our current cadential support is superficial and we would expect this to manifest in weak and contrived phrase and section boundaries.

We use no melodic shaping whatsoever. In fact aside from simple pitch constraints and basic repetition we provide no directed melodic support at all. One might expect this to result in generally worthless melodic material. Another serious shortcoming is the absence of any counter-melodic material aside from trivial thematic cannoning.

Higher level structure is currently limited to the enveloped control of global parameters and the probabilistic reuse of short thematic fragments. We expect that this should provide reasonable global structure for some parameters, harmonic tension, dynamic and orchestration for example, but provide no value for other structural issues, such as repetitions and variations that require more than local memory.

We currently provide no mechanism for escaping beyond the bounds of our preset accompaniment patterns. Furthermore we are currently limited to only two types of accompaniment pattern. We anticipate that this will severely limit the range of stylistic output that the system is capable of, but is not difficult to extend.

Results and Future Work

Here we outline our subjective reactions to the musical output of our system (data) and sincerely hope that interested readers will also review the musical examples made available here:
http://impromptu.moso.com.au/acmc08_examples.html

Listening back to the generated music we were somewhat surprised by the amount of local cohesion. With only a few simple operations, such as thematic repetition and re-use of root progressions we feel the music has reasonable local structure without being overly restrictive. Most importantly we feel that our harmonic implementation is currently operating well with a reasonable balance between harmonic novelty and structural cohesion. Overall we are happy with our current implementation of Hindemith's chord quality system. We do however need to make improvements to the sophistication of our cadential treatment in order to help clarify phrase boundaries and to help drive the music at section borders. This indicates weaknesses in our current to root-progression generation.

When listening to the music generated by our system there are many obvious deficiencies. Most pronounced is the lack of meta structure. This is hardly surprising as it is a common complaint about generative musics and our own shortcomings section anticipated this poor result. We are not overly concerned at this stage however, as we hope that introducing a "memory" into our system will help to introduce opportunities for global cohesion, such as large scale formal repetitions and more formalised thematic developments. We are also working with various user interface tools to allow higher level human control over parameter automation and structural boundaries that should provide significant assistance. The development of these interfaces is directly in line with our desire to produce usable tools for working professionals in film, TV and computer game environments, and early prototypes of these interfaces are discussed in an accompanying publication.

Our voice leading, as we speculated earlier, does indeed cause some concern, although not as significant as we anticipated. Our concerns over common voice leading issues such as parallel movement do not appear to cause significant problems. In general we have no major complaints about the shortest path approach which provides relatively smooth voice movement. Of greater concern however is the arbitrary wandering of this movement, which can lead to voice bunching and unnecessary voice crossover. We will need to address this by forcing some form of inner voice boundary checking and possible use of medium scale melodic contour control.

Our minimal set of accompaniment styles is also a severe limitation on the variety of output possible from the current system. Indeed our requirement that accompaniments be somewhat predefined is an inherent weakness in the system more generally. In future work we would like to investigate extending the instrument agent definition to provide some form of environmental listening providing each instrument agent with the ability to perceive it's relationship to surrounding material. Our hope is that we may be able to provide automatic accompaniment based around various gestalt principles of grouping, symmetry and self similarity.

One of the great surprises for the work so far is how well completely random thematic material can work. This is not to say that it is reliable, indeed random thematic materials can be as bad as they can be good. However, given that our thematic implementation is a placeholder only, we find its operation surprisingly serviceable. Our intention is to extend this current system to provide greater local scope. For example, we currently only support short thematic fragments, we would like to support longer phrases of melodic material which could also be sub-divided into smaller thematic components. This would provide us with the interesting expositional material that we currently enjoy in the systems output but would also provide longer more cohesive melodic passages. This would also provide us with more cohesive phrase level operation, a current deficiency linked to the aforementioned cadential weakness.

This brings us to our most significant insight. Overall we are convinced that the success of the system to date is very largely due to our holistic approach to implementation. In other words, we feel that the output of our present system is far greater than the sum of its often greatly limited parts. This is in line with our initial expectations, and on the surface would appear to support our methodology. However, we are increasingly concerned that a byproduct of this success is a severe limitation to the range of aesthetic output the system is capable of. In other words, we are concerned that our approach maybe producing a single "work" rather than a system capable of producing significantly varied "works". At this early stage it is difficult to know if the system's aesthetic limitations are due to its partial implementation or whether there is a more serious methodological problem with our approach. We will need to keep this strongly in mind while continuing to develop the system.

Conclusion

This has been an introduction to the ongoing work at the Australian CRC for Interaction Design into the generation of orchestral music.

Acknowledgments

The work reported in this paper has been supported by the Australasian CRC for Interaction Design (ACID) through the Cooperative Research Centre Program of the Australian Government's Department of Education, Science and Training. .

References

Brown, A. R. (2004). An aesthetic comparison of rule-based and genetic algorithms for generating melodies. *Organised Sound*, 9(2): 191-198.

Brown, A. R. (2005). Exploring Rhythmic Automata. *Proceedings of the Applications of Evolutionary Computing: EvoWorkshops 2005*, Lausanne, Switzerland. Springer, pp. 551-556.

Brown, A. R. (2005a). Generative Music in Live Performance. *Proceedings of the Australasian Computer Music Conference*, Brisbane, Australia. ACMA, pp. 23-26.

Brown, A. R., Wooller, R. and Miranda, E. R. (2008). Interactive Evolutionary Morphing as a Music Composition Strategy. In, *Music As It Could Be: New Musical Worlds from Artificial Life*. E. R. Miranda, Ed. Madison, Winconsin, A&R Editions.

Clark, J. W. (2007). *Hindemith's Analyses in The Craft of Musical Composition*. Retrieved 7 April, 2008, from http://jasonwclark.com/hindemith.aspx.

Clynes, M. (1984) The Secret Life Of Music, *In Proceedings of the 1984 International Computer Music Conference.* San Francisco ICMA

Cope, D. (1992). Computer Modelling of Musical Intelligence in EMI. *Computer Music Journal*, 16(2): 69-83.

Cope, D. (2005) *Computer Models of Machine Creativity* Cambridge, MA: MIT Press. 474 pp.

Cope, D. and Hofstadter, D. R. (2001). *Virtual music : computer synthesis of musical style*. Cambridge, Mass., MIT Press.

Crocker, R. (1966) *A History Of Musical Style* McGraw-Hill NY

Desain, P. & Honing, H (1993) Tempo Curves Considered Harmful. *In Time and Contemporary Musical Thought*. J.D.Kramer (ed.) Contemporary Music Review 7(2)

Hömel, D. (2004). ChordNet: Learning and Producing Voice Leading with Neural Networks and Dynamic Programming. *The Journal of New Music Research*, 33(4): 387-397.

Huron, D. (2001). Tone and Voice: A Derivation of the Rules of Voice-Leading from Perceptual Principles. *Music Perception*, 19(1): 1-64.

Ponsford, D., Wiggins, G. and Mellish, C. (1999). Statistical learning of harmonic movement. *The Journal of New Music Research*, 28(2): 150-177.

Sorensen, A. (1998). A Markov Approach to Style Replication. *MA Thesis QUT*.

Sorensen, A. (2005). Impromptu: An interactive programming environment for composition and performance. *Proceedings of the Australasian Computer Music Conference* 2005, Brisbane. ACMA, pp. 149-153.

Sorensen, A. and Brown, A. R. (2007). aa-cell in practice: an approach to musical live coding. *Proceedings of the International Computer Music Conference*, Copenhagen. ICMA.

Towsey, M., Brown, A. R., Wright, S. and Diederich, J. (2000). Towards Melodic Extension Using Genetic Algorithms. *Proceedings of the Interfaces: The Australasian Computer Music Conference*, Brisbane. Australasian Computer Music Association, pp. 85-91.