Brown, A. R. and Kerr, T. 2009. Adaptive Music Techniques. Improvise: The Australasian Computer Music Conference, Brisbane, Australia: pp. 26-31.

ADAPTIVE MUSIC TECHNIQUES

Andrew R. Brown
Creative Industries Faculty,
Queensland University of
Technology

Thorin Kerr
Australasian Cooperative
Research Centre for Interaction
Design

ABSTRACT

There are many interactive media systems, including computer games and media art works, in which it is desirable for music to vary in response to changes in the environment. In this paper we will outline a range of algorithmic techniques that enable music to adapt to such changes, taking into account the need for the music to vary in its expressiveness or mood while remaining coherent and recognisable. We will discuss the approaches which we have arrived at after experience in a range of adaptive music systems over recent years, and draw upon these experiences to inform discussion of relevant considerations and to illustrate the techniques and their effect.

1. INTRODUCTION

Adaptive Music reorganises existing musical material so as to produce new musical experiences based on that material. Adaptive music employs many techniques similar to those used by notational composers to conduct musical variation. Therefore our techniques for adaptation work with familiar and recognisable musical concepts such as pitches and rhythms and operate, primarily, on symbolic representations of music. This is significantly different to Adaptive Audio systems which largely rely on the scheduling of audio recordings and and manipulation of the sound by digital signal processing. An early example of adaptive music systems is the iMuse engine developed in the 1990s for use in video games by LucasArts [1].

Our Adaptive Music techniques have been implemented as algorithms which manipulate symbolic representations of music and respond to instructions in real-time. This gives rise to many possibilities for developing interactive performance environments and collaborative musical experiences involving interaction with symbolic musical material.

Music production environments have typically offered relatively basic interaction with symbolic musical material. MIDI sequencers, for example, typically offer editing facilities on individual notes, or groups of notes in ways which have little to do with actual musical transformations. Notational packages can offer useful manipulation techniques, although the design of these tools revolves around expediting note level composition (to printed score), rather than re-interpreting existing work.

2. BACKGROUND

There are a variety of uses for adaptive music. Our motivation was the provision of interactive music jamming software for musically inexperienced users; typically children or the disabled. Adaptive music techniques are also ideal for artistic installations where the musical material needs to respond to user interaction or to changes in the environment. One of the most ideal uses of adaptive music is in computer games because these are entirely digital environments that are inherently interactive and involve unpredictable real-time state change. The techniques have been used as early at 1987 by Toshio Iwai in the *Otocky* game by Famicom [2]. More recently, the Electronic Arts game *Spore* uses adaptive music extensively, for example varying the instrumentation and rhythmic density of parts in response to game state changes.

There have been a number of approaches to real-time adaptive music using symbolic representations that are worth reviewing by way of contextualising the techniques described in this paper. Interestingly a number of these have been associated with commercial software applications. These include, M and JamFactory [3], DirectMusic Producer [4], Karma [5], MadPlayer [6], Notakle [7], Synfire Pro [8]. We are not including in this review more general-purpose music environments, such as Max/MSP that can be used for creating adaptive music systems, nor are we including generative systems, such as Band-In-A-Box, that tend not to be interactive or adaptive. There are a range of algorithmic techniques used across these systems.

2.1 Templates

The provision of a harmonic template can go a long way to providing a musical backbone for generated music. In a similar way that chord progressions are prescribed in Band-In-A-Box, chord progression templates can be declared in DirectMusic as

ChordMaps, and in SynFire harmonic progressions provide a structural basis for surface level changes to phase materials. Structural templates can also be used to dictate large scale form. Often templates are manually described but can also emerge algorithmically as the music progresses as described for harmonic material by Sorensen and Brown [9].

2.2 Abstraction

The uses of abstraction as a method of isolating musical structure has a long history, evident in the analytical musicology of Schenkerian analysis [10] and the Generative Theory of Tonal Music [11]. While most real-time music systems are somewhat limited in their ability to describe, create or even track hierarchical abstractions (with a few exceptions [12]), some adaptive music systems use abstraction as a means of generalisation to inform generated variations in surface level output. For example, Cogitone uses gesture contours to describe musical phrases. Contours can be manipulated by process and by hand. Mappings to and from these contours are not necessarily deterministic and surface details reflect the current context (such as key and metre) when rendered at run time. Another abstraction often used are pitch set and series. For example, M uses lists of pitches, rhythms, and dynamics that can be looped/phased and reordered. Noatikl supports the creation of pitch class sets (scales) and rhythm sets and allows manipulation of materials based on those.

2.3 Recombination

Variation can be provided by selecting from alternative tracks or patterns. This technique is used within DirectMusic's 'style' data type that can contain track alternatives that can be selected based on some state variable or interactive controller. Timbral recombination allows for the re-voicing of musical tracks, and the MadPlayer makes significant use of this such that generated tracks can select from a range of alternatives for each instrument part.

2.4 Transformation

A long-established composition technique is to modify a motif by transposition, arpeggiation, expansion, contraction, inversion, and so on. A number of systems have employed these techniques to varying degrees including DirectMusic, Cogitone and Noatikl. The Karma system makes much of these kinds of transformations, often based around arpeggiation, as a way of elaborating on the musical input (such as a chord or phrase).

2.5 Probability

The most common functions used to control probability are random selection and markov tables. These processes have been part of computer music composition since the computer-assisted composition of the 'Illiac Suite' [13]. David Zicarelli made extensive use of probability in M and of Markov process in JamFactory. While probability is an effective an effective real-time technique, widely exploited in the Koan system for example and still a significant technique in Noatikl, the use of Markov chains requires analysis of data that can limit its responsiveness to change, but in the right context such as used interactively in The Continuator [14], can be quite effective and efficient.

In this paper we demonstrate the techniques which we have found useful in developing adaptive musical systems in performance-oriented environments.

3. MUSICAL ELEMENTS AND MANIPULATION

Music theory identifies numerous musical concepts which arise though common techniques employed by composers to organise musical material. Thus we have concepts which exist beyond individual notes such as melody, harmony, rhythms, or even form. These techniques have evolved as elements of compositional practice, however it is also worthwhile considering other musical elements which aren't expressly identified as compositional techniques.

George Pratt [15] in his work on educating music perception defines a relatively wide range of music elements that include obvious ones, like pitch and rhythm, but also less traditional ones such as spatialisation. Music perception identifies elements of music that are significant in effecting people's responses to music. Some of these are traditionally 'compositional' parameters and other 'performative.' Amongst the most commonly cited are tempo, key, and use of rubato [16].

Our criteria for selecting elements for our adaptive techniques includes that they must be;

- able to be performed in real-time (a logical and computational constraint),
- robust across a range of musical works and styles,
- make a significant perceptual difference to the musical output.

The elements of music that are are available for adaption is also constrained by the parameters in the music representation system being employed.

4. MUSICAL REPRESENTATIONS

Horacio Vaggione has observed that “there is no musical composition process (instrumental, electroacoustic, or otherwise) without representational systems at work [17 p.58]. 'At work' here implies that the representation system in use carries with it, and influences, a mode of musical understanding. Some computer-based music representation systems draw from Common Practice Notation, such as MIDI or Csound scores. Others reflect a signal processing mentality as applied to musical parameters (MAX/MSP, PD). Some other kinds can include graphic scores (UPIC) or piano-roll views. Suffice to say, the representation of musical elements in a software system becomes an important factor to the understanding of music and musical transformations adopted by the software user. In our work we are using a MIDI files of compositional material as the initial material. This means that the immediate representation makes available elements such as event onset and end times, chromatic pitch, velocity, instrument number, and so on.

5. TECHNIQUES FOR MUSICAL ADAPTATION

The techniques described here are those we have found useful over a range of projects, in particular we have implemented them in an algorithmic music application for the XO Laptop designed and distributed by the One Laptop Per Child Program, which we will defer detailed discussion of to a forthcoming publication. Suffice to say that the system relies on MIDI data as the basic material for a musical description that is performed on a sample-based synthesizer.

Based on the material provided in the MIDI file, adaptive algorithms conduct significant symbolic manipulation of the musical data. As will be seen, there is comparatively minimal adaptation of audio synthesis or rendering parameters. By way of example the bass line in figure 1 will be used as a basis for transformation examples in this section.

The transformation of each musical element is discussed below.

5.1 Pitch Range

A common and simple technique to affect pitch material is through transposition. Key modulation within a work can be used to produce an increase in tension, or a resolution to the form of a work. However, beyond this scope the effectiveness of transposition proves relatively limited. Furthermore, novice users have difficulty using transposition in musically sensible ways.

Our approach to pitch transformation concentrates instead on the pitch range of a part. We have found that this provides comprehendable and controllable variation. The basic technique is to expand or contract the range covered by the pitches of a part. This is equivalent to the stretching or compression of the pitch contour. Such range variation requires some constraints. A useful pivot pitch, relative to which the range is calculated, which can be a root pitch either at or near the bottom of the original pitch range.

When the range is at its minimal level, the phrase contour is a flat line. All notes in the phrase have the root pitch. There is theoretically no limit to the amount of expansion, but practical limits include the playable range of the instrument, and what expansion amount is musically sensible. The resultant expansion is dependent on the original pitch range of phrase, but in our experience an expansion multiplying the original phrase range between 0 - 2 times is generally sufficient. Figure 2 shows the example with expanded pitch range.

A final constraint is to quantise the resultant pitch to a given pitch set. Effective approaches include the use of a pitch class based on the pitches used in the original phrase, or an appropriate scale or mode for diatonic music.

Percussion instruments respond differently to pitch control. Selecting different 'notes' in a MIDI drum kit, or alternatively, adjusting the audio playback pitch of percussive sounds doesn't provide conventionally musical results. Instead, we have taken the approach of adjusting the selection of drum instruments already in use by the music. An expanded pitch range then takes advantage of the entire drum-kit range. A reduced range will concentrate the drum selection toward a particular drum sound, typically a bass drum.

5.2 Rhythmic Density Thinning

A significant contributor to the 'mood' or 'energy' of the work is the frequency of events. Put more musically, this equates to the number of notes per time period which we call the rhythmic density (not to be confused with the textural density which equates to the number of concurrent independent parts).

Rhythmic density thinning takes the approach that the given piece of music is the 'most dense' version of the music. Assuming the music conforms to regular metric structures, the rhythmic density can be reduced by removing notes that occur in less-
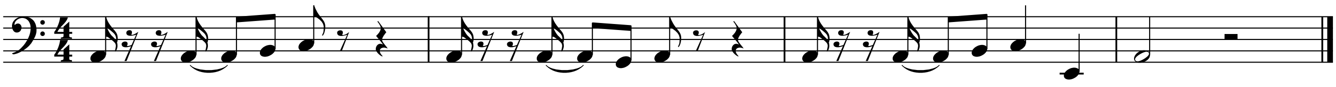
Figure 1. The original musical phrase.
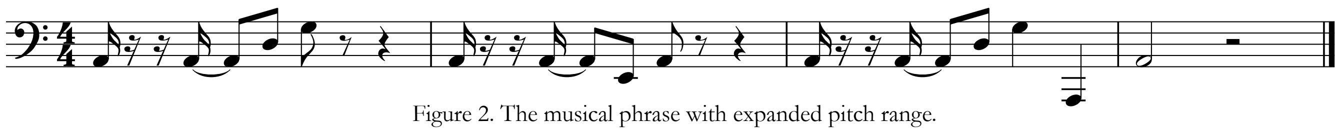
Figure 2. The musical phrase with expanded pitch range.

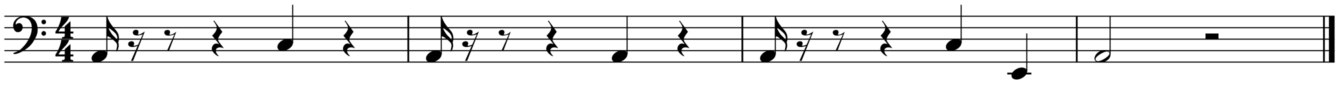

Figure 3. The musical phrase with reduced rhythmic density.

dominant metric positions. For example, in simple quadruple time (4/4) we assume a hierarchy dominance beginning with beat1, then 3, then 2 and 4 followed by further subdivisions of the beat which follow a similar hierarchy.

Events (notes) are then filtered based on their metric position, and where that position sits along a continuum from least to most dominant. Figure 3 shows an the example phrase with a reduced rhythmic density.

While this technique in its raw form is effective, we have found that the musical change from stage to stage along the filtering continuum can be too sudden. To compensate for this effect we soften the change from stage to stage by implementing a 'porous' filter that 'lets through' a number of notes that would otherwise be filtered. Another side effect of this filtering is that note durations need to be adjusted (lengthened) to compensate for the 'space' created by deleted notes. This duration operation occurs independently of other processes which affect the duration of notes (see Articulation below).

5.3 Articulation

In musical performance the articulation of a note can mean a number of things including tuning, attack, dynamic level and so on. In this situation we use articulation in a more limited sense to simply mean the performed duration of each note. Whether a phrase is played legato (sustained notes) or staccato (abbreviated notes) can make quite a difference to the 'mood' of the music. The algorithmic aspect of this is quite trivial, to scale the note duration by some articulation factor. This factor when set at 1.0 will playback the note as 'recorded' in the MIDI file (or as adjusted by some other algorithmic process such as rhythmic density), a factor near zero will produce the shortest note length while factors greater than 1.0 will extend the note. Given that most human performances have note lengths around 80% of the written duration, a factor of 1.2 or more will start to produce overlapping notes, creating an effect not unlike holding the sustain pedal down on a piano.

5.4 Dynamic Level

The dynamic level in musical performance is largely a matter of loudness, but is also associated with timbre, attack and other factors that correspond with more energetic playing of an instrument. In implementing this technique we confine ourselves to a simplistic notion of dynamic as loudness. Of course given that we use MIDI velocity as a carrier of the dynamic value it is possible to use a synthesizer which responds with more sophistication than simple volume adjustment. Similarly, all scaling in our examples are linear and we leave it to the synthesizer to convert these into the logarithmic loudness curve typical of musical instruments.

We employ two dynamic controls, ‘overall’ and ‘pulse dynamic’ variations: Our overall adaptive dynamics simply scale the existing dynamic value of each note from the MIDI file. Given that most performance in MIDI files average at around 70-80% of the available dynamic range some capacity to increase dynamic level can be expected within the limits of legal parameter ranges

Pulse-based dynamic change mimics the way performers add emphasis to certain beats in regular metrical music. We use a simple periodic function (summed cosines) to imitate this emphasis. The frequency of the periodic function needs to match the pulse or beat duration of the current metre. Controlling this function means controlling the amount of emphasis given to accented beats. Increasing emphasis, increases accents on certain beats. Note that a setting specifying no emphasis will provide no additional emphasis on beats, however accents recorded in the original performance are retained.

5.5 Tempo

The speed of music has been shown to be a significant contributor to musical affect. Adjusting the tempo involves a relatively trivial change in playback speed. As with the dynamic level, research has shown that quasi-periodic variations in tempo are common in human musical performances [18, 19]. To imitate this and provide adaptive control over it we implement a subtle periodic tempo change, again using summed cosine functions, that approximates this effect. Here the style of the music is important: For example, the effect is quite effective when applied to Western music of the Romantic period which utilises rubato quite liberally. It is less effective when applied to modern or electronic music styles with rigid tempos.

5.6 Timbre

Timbre manipulation is not the focus of adaptive music techniques, however some simple timbre manipulation is provided. In particular, a resonant filter is provided which is mostly effective for the manipulation of electronic styles of music. Filter cutoff is the parameter adjustable by the user. A useful amount of resonance can be calculated as a function of cutoff, peaking at a mid range of between 700 to 1000 hz, and tapering at audible extremes. If a resonant filter is not suitable for the style of music, or the instrument, then the quality of the sound can be changed by selecting an alternate instrumentation for the part.

There are audio examples of all these techniques available online at http://www.explodingart.com/acmc2009/.

6. INTERACTION, CONTROL AND MAPPING

There are a variety of ways to represent changes in musical parameters through an interface. On the XO Laptop users control the amount of musical transformation by controlling the spacial position of an instrument. Movement along each axis adjusts a particular group of musical transformations. Here users understand the transformations in terms of a 'stylistic' change to the music. A similar approach is provided with the Jam2Jam network jamming software on the Macintosh. In this case users are given access to the specific musical parameters along each axis. With this model, users gain an understanding of each individual transformation, and how each transformation contributes to overall stylistic qualities. Interaction can also be collaborative through the use of supplementary controllers such as MIDI controllers or iphone applications, or with networking features available in the software.

7. TESTING AND EVALUATION

These algorithmic processes have been developed through field trials with several systems, in particular four iterations of the Jam2jam network jamming software that have been tested in numerous locations around the world [20, 21]. Observations provided by participants in the trials has guided the direction of development of the algorithms and the mechanisms for interaction with the algorithms. Of key importance was gaining an understanding of the relationship between the users experience, the mechanism for interaction and the effectiveness of the algorithms. For example, through trials we found that, although the interface was easy to interact with, this did not always translate into understanding musical concepts in play, as illustrated through this observation from a teacher:

> "when being used with students, some training needs to occur. Although the software is intended to be experimented with, students find it confusing to just pick up and use. They need to have it demonstrated so they know what it's possibilities are" (Personal correspondence, 2009).

In observations like this, it is unclear whether modifying the algorithms or the interface will improve the experience. In fact, the solution to this feedback, and others like it, was to change the representation of a musical parameter. Rather than simply providing an interface with controls to a 'black box' musical application, a user could visually see settings on the screen which corresponded with the actual musical parameter settings.

As well as observations of users interactions with these systems that have informed the effectiveness and significance of these changes, the various jam2jam iterations have involved three re-implementations using different programming languages (Java, Scheme, and Python) and music systems (jMusic, Impromptu, and Csound). This process has also refined the way in which musical adaptations can be most efficiently be represented as algorithms.

8. SUMMARY

Adaptive music techniques offer a means to enhance the expressive terrain of existing music by translating musical concepts into performable parameters. In the course of this research we have found that the representation and effectiveness with which to interact with musical concepts is crucial in enabling creative engagement with those concepts. This paper has presented some of the techniques, and refinements of those techniques, which have proved effective.

Audio examples of all the techniques described in this paper are available online:

http://www.explodingart.com/acmc2009/

9. ACKNOWLEDGMENT

This research was supported by the Australasian CRC for Interaction Design, which is established and supported under the Australian Government's Cooperative Research Centres Program.

10. REFERENCES

[1] Land, M. and P. McConnell, *iMUSE*. 1994: Berkeley.

[2] Collins, K., *An Introduction to Procedural Music in Video Games.* Contemporary Music Review, 2009. **28**(1): p. 5-15.

[3] Zicarelli, D., *M and Jam Factory.* Computer Music Journal, 1987. **11**(4): p. 13-29.

[4] Fay, T., *DirectMusic Producer.* 1998, Microsoft Corporation: Redmond.

[5] Kay, S., *Karma*. 2001, Korg USA: Melville, NY.

[6] MadWaves, *MadPlayer.* 2004, MadWaves: London.

[7] Cole, T. and P. Cole, *Noatikl.* 2008, Intermorphic: London.

[8] Schnoor, A., *Synfire Pro.* 2008, Cognitone: Hamburg.

[9] Sorensen, A. and A.R. Brown, *A Computational Model For The Generation Of Orchestral Music In The Germanic Symphonic Tradition: A progress report*, in *Sound : Space - The Australasian Computer Music Conference*, S. Wilkie and A. Hood, Editors. 2008, ACMA: Sydney. p. 78-84.

[10] Schenker, H., *Harmony.* 1980, Chicago: University Of Chicago Press.

[11] Lerdahl, F. and R. Jackendoff, *A Generative Theory of Tonal Music.* 1983, Cambridge, MA: MIT Press.

[12] Gifford, T. and A.R. Brown, *Automated Improvisation*, in *Australasian Computer Music Conference.* 2009, ACMA: Brisbane. p. (in this volume).

[13] Hiller, L.A. and L.M. Isaacson, *Musical Composition with a High-Speed Digital Computer.* Journal of the Audio Engineering Society, 1958. **6**(3): p. 154-160.

[14] Pachet, F., *The Continuator: Musical Interaction with Style*, in *International Computer Music Conference.* 2002, ICMA: Göteborg, Sweden. p. 211 - 218.

[15] Pratt, G., *Aural Awareness: Principles and Practice.* 1990, Buckingham: Open University Press.

[16] Livingstone, S.R., et al., *Controlling Musical Emotionality: An Affective Computational Architecture for Influencing Musical Emotions.* Digital Creativity, 2007. **18**(1): p. 43-54.

[17] Vaggione, H., *Some Ontological Remarks about Music Composition Processes.* Computer Music Journal, 2001. **25**(1): p. 54-61.

[18] Juslin, P.N. and J.A. Sloboda, *Music and Emotion: Theory and research.* 2001, Oxford: Oxford University Press.

[19] Friberg, A., *pDM: An expressive sequencer with real-time control of the KTH music-performance rules.* Computer Music Journal, 2006. **30**: p. 37-48.

[20] Dillon, S., et al., *Communities of sound: generative music making and virtual ensembles.* International Journal of Community Music, 2008.

[21] Brown, A.R. and S. Dillon, *Networked Improvisational Musical Environments: Learning through online collaborative music making*, in *Music Education with Digital Technology*, J. Finney and P. Burnard, Editors. 2007, Continuum: London. p. 96-106.