1 Introduction

1.1 Background motivations

The delivery of electronic music rarely operates in the same spontaneous way as acoustic music and, in contexts where some form of adaptivity exists, the aesthetics are restricted and shaped by technological limitations. **It is the goal of this research to investigate techniques that both enable greater adaptivity and provide a new set of aesthetic possibilities for mainstream electronic music delivery.** “Enabling adaptivity” can be defined and evaluated as a technical problem, while “providing a new set of aesthetic possibilities” is an artistic problem and thus more exploratory in nature. Consequently, my research method combines technical development and reflective exploration with both informal and formal evaluations. In particular, the goal was pursued through the investigation of note sequence morphing, where a hybrid transition is generated between source and target note sequences. The musical genre within which the research operates is Mainstream Electronic Music (MEM), which I define as the popular and largely instrumental form of electronic music constructed from loops of layers such as drums, bass, lead, chords and sound effects. While the techniques of note level morphing I developed could be applied to virtually any MEM context, particular niche areas I have identified that could reap substantial benefit include: live MEM, computer game music and interactive installations.

1.1.1 Live MEM

The first context is the live delivery of MEM music through a DJ or by using live music software. The skilled DJ/producer gauges the mood and taste of the audience and selects appropriate tracks to mix. However, despite the range of aesthetic possibilities within pre-produced music, the technology for mixing is usually limited to sonic effects such as equalization, cross-fade, delay, time/pitch stretching and others. Unless the music has been carefully selected to ensure a degree of compatibility, the transitions can sound extremely awkward. It is particularly difficult to create an intelligible extended mix that might be perceived as ‘new’ hybrid music by the audience.

A legacy of technological limitations and affordances have forced aesthetic limitations on the live delivery of MEM that have come to be accepted by the audience as part of the musical form within which skilled practitioners operate. This is most obvious in Electronic Dance Music (EDM) where live mixing of tracks is common. Specific aesthetic features are that tracks are compatible; time-scales are long; sonic interest is more important than pitch and melody; tempo is consistent during mixes; and rhythmic repetition is more appropriate than varied phrasing. All of these and other aesthetic limitations can be correlated, at least partially, to limitations of technology – tracks must be compatible because the current tools do not afford on-the-fly compositional changes and the pre-production of customised transition material is uneconomic and unfeasible if track selection is to occur in a truly adaptive fashion. Time scales are long because the change in mood between tracks proceeds in small increments. Change in timbre is important because of the availability of realtime-manipulable (and thus adaptive when used by a sensitive human) sonic effects in audio production and mixing. Tempo is consistent during mixes because it is extremely difficult to beat-match and increase the tempo on two turntables simultaneously.

Considering the connection between the technology and aesthetics in adaptive electronic dance music delivery, new software for hybridising and transitioning between electronic dance music tracks could allow a range of new aesthetics to bloom. The ability of morphing to hybridise even very divergent tracks could enable new approaches to track selection, with tracks that would otherwise not be heard back-to-back becoming compatible. With smoother morphs, dramatic changes may become feasible over shorter periods of time. Due to the use of note sequences rather than audio tracks, transitions based on melodic and harmonic techniques could be explored, rather than simply equalisation, cross-fading and effects.

1.1.2 Computer games and multimedia

A second application for note level morphing is in computer games and multimedia, where the small amount of music, relative to the time spent playing the game or navigating the media, usually ensures that the music will become repetitive and boring at some stage (Sanger 2004). More importantly, although musical changes occur in response to narrative cues, the material must be cleverly composed so that each piece is compatible with every other that borders it (Apple 2006), constraining the musical possibilities substantially. The connection of divergent musical themes is very awkward without customised transitional material, which is uneconomical. Consider the following: a game with only 20 different states, but that requires a transition from each state to every other, would necessitate 190 transitions – too much material to produce economically and increasing exponentially with larger projects. Therefore, techniques that enable the automatic creation of such transitions would make composition processes more efficient and currently intractable game music projects possible. While some computer game music tools, such as Direct Music Producer (Fay et al, 2003), enable more efficient organisation of transitional material, a tool that exploits sophisticated algorithmic composition techniques is yet to become widely available. As well as transitions, hybridisation techniques could be used to create a huge number of 'mutant' pieces, thus increasing the amount of music that is able to be applied within the game overall and reducing repetition.

1.1.3 Interactive installations

A third context for note sequence morphing is music for interactive installations that are displayed in locations such as art exhibitions, conferences, museums, theme parks and festivals. Social installation interfaces such as tabletop marker tracking systems are particularly relevant to morphing, which can be easily projected onto a topology. With only a single parameter to define the position between the source and target, note sequence morphing is simple to use for the general public. As well as this, if the source and target music are divergent, a large range of musical possibilities are nonetheless available. The potential number of musical states can be increased significantly when layers of the music – such as bass, drums and lead – are morphed independently.

1.1.4 Additional motivating contexts

Other adaptive and electronic music contexts such as accessible music therapy (Cost-287 2007) and computer assisted composition could benefit in ways similar to live MEM, computer games and interactive installations. The selection of MEM as the primary musical genre is partially motivated by the potential economic benefits of working in a popular genre, as well as a personal interest in the music. Primary motivations for extending the current knowledge of note level morphing are summarised thus:

- Increased adaptivity in the delivery of MEM.
- Increasing the spectrum of aesthetic possibilities for musical transitions.
- Increased efficiency in MEM composition, both adaptive and otherwise.
- Increasing the possibilities and affordances for musical hybridisation.

Note level morphing is highly relevant and applicable to the research goals of enabling greater adaptivity and providing a new set of aesthetic possibilities for MEM delivery.

1.2 What is note sequence morphing?

Note sequence morphing is the task of integrating two separate note sequences to create a hybrid that may work as a transition between the originals. The 'note sequence' aspect indicates that the algorithm operates on musical notes, with attributes such as pitch and onset, rather than sound waveforms or graphics.

Larry Polansky's morphing definition (1992) can be simplified and adapted to the specific context of note level morphing as follows. A morphing algorithm, denoted by f, is limited to two musical inputs. One of these is the **source** music, denoted by S, and the other is the **target** music, denoted by T. For example, S might sound like sound example (~1.1), and T might sound like (~1.3). The output of the morphing algorithm is the **morph** music, and is denoted by M. M is a kind of hybrid transition between S and T, which, continuing the examples, could sound like (~1.2)[1]. The influence of S or T in M is controlled by the **morph index**, or Ω, which is typically a user-specified weighting. Ω is normalised such that when $\Omega = 0$, $M = S$ and when $\Omega = 1$, $M = T$. We have $f(S, T, \Omega) = M$.

The three note level morphing algorithms I created all fit this general definition. The various approaches differ substantially, as explained in chapters five, six and seven, each of which deal with a different morphing algorithm.

Two artistic purposes for the use of morphing in music can be distinguished. One is to create a **transition** whereby S, M and T can be positioned in sequence to create a smooth and/or coherent whole. This has potential in multimedia, computer games, live DJ mixing or any context where automatic transitioning between music is required. Another artistic purpose can be to create a **mutant** or **hybrid**, whereby M has aesthetic interest in the way it shares properties of S and T, but may or may not function as a transition. This is more applicable to computer assisted composition and musical experimentation. Despite the distinction, these two purposes often overlap considerably, especially when considering a lengthy transition that can operate as an independent piece of music.

[1] The morph example was produced by the *LEMorpheus* software I developed through this research.

1.3 Compositional and philosophical issues related to morphing

Various compositional and philosophical issues related to morphing require some introductory treatment, including: the difference between interpolation and morphing, n-source morphing, similarity measures, continuous versus discrete changes during transitions and the abstract description provided by the morph-index.

From discussions with various peers (Collins 2006; McCormack 2006), I have found it important to distinguish between **interpolation** and **morphing**. Interpolation is a technique for estimating unknown values between known points. Morphing, however, is more concerned with aesthetic integration of separate patterns. The technique of interpolation may feature to different degrees, or even not at all.

While the style of morphing discussed within this research is purely between source and target, morphing can also occur between multiple sources, which is called **n-source morphing** (Oppenheim 1995). There are some musical implications to this. In particular, using n-sources can shift the focus to be less on the transition and more hybridity and exploration of the much larger "morph space". Simple source-target style morphing is easily applied to a musical work by mapping the morph-index to time. In contrast, n-source morphing is more complex, because the number of dimensions describing the morph space is increased, while the limitation of the "time" dimension in the musical work remains. That is, it is inevitable that one must consider how to navigate the morph-space if one is to generate a musical work, which in turn makes exploration more important. Because the generation of an N-source transition is thus more complex than with source-target morphing, and because multiple sources increase the likelihood that unique and possibly interesting combinations will result, N-source transitions also afford the generation of new, hybrid musical states, as opposed to the transitions between states.

Similarity measurement is pertinent, as it could be considered a kind of inverse function to morphing. That is, while morphing generates output at a similarity level specified by the morph-index, similarity measurements analyse two inputs to determine a level of similarity between them. Similarity measurements can be usefully applied to morphing algorithms to track the progress of the morph (Horner and Goldberg 1991), or they can adapted for use in reverse (Polansky 1992). If the similarity measurements are metric, that is, upholding the property of triangle inequality, they can be used to define a space, and morphing can proceed simply through interpolation through the space (Mathews and Rosler 1969).

There are a number of techniques for determining similarity between note sequences, the most prominent of which are briefly mentioned here. One approach is to convert the note sequence into continuous envelopes of note parameters (Mathews and Rosler 1969) and take the difference between them. The problem with this is that the similarity is based on metric distance between the parameters, and this often contravenes musical notions of similarity. Note-sequences, in being non-continuous, can be perceived as strings of characters, of which there has been significant research into similarity measurement techniques (Orpen and Huron 1992). The standard approach with strings is to allow a set of operations – for example, add, delete, shift – and count how many operations it takes to convert the source into the target (Damerau 1964). The number of operations will be proportional to the level of dissimilarity. For note sequences, the set of operations can be easily adapted to a musical context (Orpen and Huron 1992; Cope 1997).

If the source and target are considerably different, it is rare for composers to attempt a continuous morph – it is much more common for changes to be introduced abruptly and transitions proceed through a series of discrete segments, for example, A to B, through C. This is not to overlook the use of foreshadowing which can indicate some quality of an upcoming change. However, when the source and target are similar, abrupt changes do not appear necessary (see the final experiment in chapter seven). In addition, variation, for example in the episodic material of a fugue (Smith 1996), is practiced widely as a way to add interest, or to generate additional “filler” material. This suggests the possibility of a similarity threshold, below which it is more natural to utilise discrete changes and above which it is easier to utilise more continuous variation. The approach taken for this research is different, in that morphing is attempted through continuous variation regardless of the similarity of source and target. The benefit of this is that new, hybrid transitions can be created where none may have been tried previously.

Finally, some philosophical consideration must be directed to the kind of abstraction that the morph index or similarity measure provides. For instance, what does “half” really mean, when the morph index is an input to a highly complex recipe which rests on analytical reductions and musical assumptions? In a practical sense, the morph index is no more than a tool to control the music. However there is an added implication that when the morph index is “half”, the result will sound equally reminiscent of both the source and target, or in the worst case, equally dissimilar to source and target. Therefore the validity of the morph-index might be cross-examined by utilizing similarity measurements. If these similarity measurements were somehow shown to produce results that were equivalent to human judgments of similarity, the intuitive understanding of the morph-index could possibly be validated. Despite this, due to the qualitative nature of music perception and interpretation, there would always remain a huge variety of different ways to interpret a morph index on “half”.

1.4 Details of the research goal

Within the broad goal of this doctorate to “research techniques that provide greater flexibility and new aesthetic possibilities for adaptive delivery of mainstream electronic music”, there are two elements of equal significance: **function** and **style**. Providing adaptivity and greater flexibility in delivery, while maintaining some degree of musical consistency, is a gain in *functionality*. Developing new and interesting aesthetic possibilities builds a *stylistic* palette, the various effects of which can be understood through user-testing.

The two artistic purposes of note level morphing can also be seen in these terms. Transitioning, although an artistic problem, is utilitarian or *functional* in nature: how to progress smoothly from source to target. Techniques for automatic transitioning have the practical outcome of enabling greater and more flexible adaptivity, in the sense that the system delivering the music is able to adapt to changes by shifting easily through repertoire as deemed appropriate, either by a user or another algorithm that analyses some kind of world state.

In contrast, hybridisation is more open-ended and related to stylistic exploration: how to integrate source and target in a way that is somehow reminiscent of both. While there are arguably as many approaches to this as there are composers, the use of algorithmic techniques provides new opportunities that are difficult to execute without a computer.

In summary, the broad goal of the research is to improve the function of adaptive MEM delivery while expanding style. More specifically, this is to be explored through note sequence morphing which simultaneously experiments with both the functional and stylistic demands of the research via the generation of hybrid transitions. With the goal thus clarified, the methods used to achieve this can now be related.

1.5 Personal background

I will now outline pertinent aspects of my personal background, so as to provide some additional context for the research. This includes musical influences, education and interests; previous studies of mine; as well as other relevant projects I have been and/or are involved in.

My musical interests began with the listening habits of my parents, which included 60s, 70s and 80s rock and mainstream electronic music. I played the trumpet briefly when I was seven and the piano and keyboard intermittently through to when I was 13. At this stage, I switched to learning classical guitar, for which I developed a mild proficiency over the next five years. At 16 I became more interested in blues, jazz and improvisatory aspects of music.

After I discovered jamming, I began to feel that memorising scores and performing them for large audiences overlooked the fun of the musical experience, both for the audience and the performer. I felt that the experience of creating music was more fulfilling than observing or performing, regardless of the skill levels involved. This view pervaded the next decade of my practice.

Towards the end of high school I began to experiment with audio editing and MIDI sequencing, teaching myself through trial and error and creating a small folio of works. Along with my guitar skills, I used these early pieces to gain entry to the music production course at QUT, for which I majored in new media.

Over the next three years of the degree, I came to adopt the computer as my primary ‘instrument’ rather than the guitar. I produced more than an album’s worth of material although my primary interest was in the spontaneous and interactive practices of computer music. I began to develop software that could respond to changes in realtime so as to control the music in a way that felt ‘live’. This was the *Live Electronic Music (LEMu)* software, which I rebuilt for my Masters project. This software was mainly designed for EDM, which had become a major interest of mine over the course of the degree, due to the influence of peers and the desire to select a relevant contemporary style of electronic music to study.

The research for this doctorate is motivated fundamentally by similar interests while being directed at techniques that are applicable to a wider range of contexts than only live electronic music. Morphing became a musical interest through the early stages of the research, when it was found to be a widely applicable and under-examined area of study.

1.6 Design of research

1.6.1 Research design context

This research seeks to model existing approaches to music and simultaneously expand on them, using technology to enable new musical possibilities. This approach is based more on engineering than analysis and as such is somewhat isolated from much of **musicology**, being more related to **computer music practice**, which is informed by **artificial intelligence**.

A useful analogy is comparing the study of bird flight and plane flight – although the plane achieves flight in a totally artificial way, we can see that building jets and observing their behaviour has taught us a lot about aerodynamics, perhaps more so, than analysing bird flight. The hope with computer music methods is that building a machine with new musical capabilities and then observing its effect will provide some new knowledge of music that is applicable to any chosen musical context.

This section will explore the research designs that have been applied to similar algorithmic music objectives in the past. This is necessary for clarifying the position of this project in relation to other research and other fields. A contextual understanding is required for any informed interpretation of the research design of this project.

There is a history in computer music of under-testing, that is, building the musical machine, *but not formally observing its effects,* a point also noted by Pearce and Wiggins (2001). Even major works are under-tested (Hiller and Isaacson 1958; Bel and Kippen 1992; Oppenheim 1995; Cope 1996; Bencina 2004; Biles 2004). This phenomenon can be attributed to the attitude of musical autonomy (Beard and Gloag 2005) that came with some traditions of classical and art music composition. With this approach, the music is considered self explanatory and in no need of interpretation. While this is acceptable in the context of composition, academic studies require more analytical rigour in order to produce clear knowledge that is able to be used and validated.

While some exceptions to the overall trend exist (Hild, Feulner et al. 1992; Mozer 1994; Phon-Amnuaisuk, Tuson et al. 1999), the issue of under-evaluation has only been explicitly examined in recent years (Pearce and Wiggins 2001; Pearce, Meredith et al. 2002). My research furthers the movement towards evaluation by including rigorous scrutiny of the musical outcomes, drawing from a combination of critical listening (Pratt 1998), survey (Mozer 1994) and automated batch testing (Pearce and Wiggins 2001). Informal techniques include listening and subjective criticism, concert performances as well as installations and proof-of-concept demonstrations.

The research design is only distantly associated with computer-assisted musicology. Schüler (2002, p 125-126) has distinguished a number of branches within this field, most of which are in some way relevant but, overall, the musicological tradition is primarily concerned with analysis, rather than automatic generation, of music. This research has involved some amount of direct musicological analysis, to define the musical context and interpret the musical results. However, much of the knowledge behind the software that I developed, which itself is considered as a rigid music formalisation, came from years of subjective experience creating and listening to music, explained above (**1.4**).

Overall, the method of algorithmic music research tends to follow the engineering pathway of iterative software development as discussed below.

1.6.2 Research through iterative software development

Although the design of this research is more closely related to engineering and computer music practice than musicology, it is nonetheless concerned with the development of knowledge concerning musical processes and their effects. This knowledge is built by the development and testing of software formalisations of music. Iterative software development as research into musical phenomena is practiced by many, but has been discussed in detail as a methodology by only a few (Brown 2007; Desain and Honing 1992).

Iterative software development can be described as repeated application of phases of: **investigation, development, evaluation,** and **presentation**. Various other descriptions of iterative software development exist and differ superficially (McConnell 1998). The important factor is not the specific title of the phases, but the fact that they are iterated many times over the course of the research. This feature sets it apart from the other major software development method, the “Waterfall Model”, criticised by Royce (1970), who invented the term. In the waterfall model, the entire project is scoped out, planned and executed in linear succession, backtracking and remodifying the plan when the results are unexpected.

As it was anticipated that the results of this research were to be unexpected, iterative development was therefore a much more appropriate model to use as a basis for research design. The phases of investigation, development, evaluation and presentation were cycled many times throughout the project and there was a significant degree of overlap, with activities often occurring in parallel. Through subsequent iterations, the amount of time dedicated to each phase shifted over time, with a focus on investigation in the beginning, moving through development, evaluation and presentation at the end. The specific details of this research process become evident in more detail as the thesis unfolds.

While iterative development was clearly a valid methodology for the research, the techniques for evaluating the musical algorithms, in terms of both music and software will now be examined in more detail. Their validity is important for the knowledge claims to be accepted.

1.6.3 Evaluation techniques

Despite the aforementioned sparsity of evaluation methodologies to follow in computer music, there are some notable exceptions (Mozer 1994; Pearce and Wiggins 2001) which have inspired the use of evaluation here. As well as this, there are a number of techniques that have been adapted from the related fields of AI, musicology, psychology and music practice which are discussed below.

Evaluating algorithmic music

Pearce and Wiggins (2001) presented a framework for evaluation drawing from AI and empirical musicology. As with this research, they view an algorithmic music system as an agent, designed to create music of particular style. This view lends itself to comparison with human compositions through empirical methods. However, my own research differs in that it is not interested in a musical '**Turing test**', the unstated objective of which is to replace the human composer. In contrast, my own emphasis is on creating tools that empower composers and producers, rather than attempting to replace them. While it is important that the music composed through algorithmic means is acceptable or 'realistic' music, my primary interest is the subjective musical qualities that differentiate the composer-agent and the human composer. Qualitative information seems to me to be more pertinent, both to the goal of improving musical algorithms and for understanding the new aesthetics of computational composition. Pearce and Wiggins allude to this when discussing the possibility of "*an experiment asking for an aesthetic evaluation of a set of patterns containing machine and human composed music*" (2001: p9). My research utilises a combined approach to evaluation of the music that covers critical listening, empirical surveys, concert performance, automatic evaluation and personal subjective evaluation.

Critical listening involves conscious awareness of the subjective listening experience and examination of the music. Subjective effects are noted and there is some attempt to understand the musical causes. Critical listening is commonly employed by composers and producers to improve their works. By playing music to peers, feedback is received as to whether the musical techniques being employed are achieving an appropriate effect, and possible improvements become evident. Pratt (1998) discusses the listening experience in terms of "effects" and “effectors”. Effects are the subjective responses of the individual, and effectors are the elements of the musical surface that contribute to the subjective effects. For example, the effector “high pace” may correspond to the effect “heart racing”. Many other researchers also discuss the subjective process of listening in terms of cause and effect (Kerman and Tomlinson 2003; Beard and Gloag 2005). Huron uses qualitative responses to evaluate the qualia, that is, subjective aesthetic responses, associated with particular scale degrees (Huron 2006, p 146). Pratt’s “effectors”, or elements of music, were employed in the final questionnaire, which I describe in chapter seven. For this questionnaire, the process of critical listening was formalised and data was gathered from a range of participants.

Concerts allow computer generated music to be tested within a realistic setting, however, it is difficult to extract useful data from the audience. This is due to the fact that strangers providing feedback to the composer of the music have a tendency towards politeness, rather than criticism. As well as this, the attention that is being payed to the music may vary significantly between individuals. Nonetheless, successfully delivering the music in a realistic setting is an indicator the music is at least functional – people will usually walk out or complain if the music exceeds the boundaries of acceptability. Because of this, performance with music software has been employed by various computer music researchers as a way to add validity to their work. For example, Biles (2004) regularly performs with GenJam and the computer composed music of the “Illiac Suite” (Hiller and Isaacson 1958) has been played by human musicians at concerts. Although I employed my *LEMorpheus* software during a number of live events, the research does not rely solely on concerts for music credibility. I also developed a “focus-concert” format, which situates the directed questioning of the “focus group” within the context of a concert – this was applied to the second note level morphing algorithm that I created, as detailed in chapter six.

Automatic evaluation techniques, as advocated by Pearce (2001), can be applied in situations where a set of attributes can be clearly identified as leading to a desirable musical outcome. These attributes function as criteria for judging the effectiveness of the musical algorithm and because they are easily extracted, automatic evaluation of the music can occur. The advantage of automatic evaluation is that large amounts of data can be tested and so various statistics and trends can be inferred. The disadvantage is that, beyond the most basic rules, it is difficult to establish a set of explicit criteria that map to subjective musical effects. I used automatic evaluation in chapter seven by generating a number of morphs with random source and target material.

The questionnaires used within this research focused on obtaining subjective responses and musicological feedback from a group of individuals with musical backgrounds. This is essentially an empirical form of critical music analysis (Beard and Gloag 2005). The primary benefit of this qualitative approach is that a range of complex compositional techniques are able to be described and justified by the various respondents. This has the practical advantage that the data generated is able to be fed directly into further development of the software.

The alternative option would have been to develop a set of criteria for a ‘successful’ morph and ask the questionnaire participants to rate the morph examples quantitatively according to the criteria. I consider such an approach to be premature for the current stage of development in the field of note level morphing. This is because there are so many note level morphing techniques that remain to be conceived and explored, that any accurate quantitative evaluation for the few techniques that have been implemented will rapidly become redundant. The criteria used might also be required to shift, with new subjective responses and values associated with new techniques. This is not to suggest that the technique of quantitative assessment is flawed, however, it would be more useful in a more competitive situation. When a diverse range of techniques and systems are already implemented, quantitative evaluation of morphs according to various criteria would be practical in allowing people to select between different approaches for their particular note level morphing application.

The qualitative approach I took appears to be fairly unique within empirical musicology, which is predominantly quantitative, with qualitative techniques typically applied to social aspects (Clarke 2004, p 92). Despite this, common issues were considered, such as the chain of interpretation, the background of the participants, the statistical significance of the data, benchmarks, realism and controlling factors that could influence the outcome of the questionnaire (Clarke and Cook 2004). I chose participants with strong musical backgrounds, which qualified them to interpret their own subjective response. This was useful in linking between musical cause and subjective effect. I provided an additional level of interpretation by summarising and analysing their comments. Because the participant responses included detailed justifications, statistically significant sample sizes were less important than if the quantitative approach had been taken. Benchmark comparisons were used in all tests, from simple cross-fades to live mixes by a professional DJ, to pre-produced morphs by a human composer/producer. A real world context (Windsor 2004, 197) was attempted for one questionnaire, while the subsequent questionnaire was conducted online to allow greater control and publicity.

Composers often utilise an informal process of listening to their own work and improving it rapidly, based only on subjective observation. This is also often applied to the development of algorithmic music software. The advantage with this approach is the ease with which personal judgements are formed, compared to time consuming formal evaluation techniques. The disadvantage is that no substantial claims can be made as to the wider applicability of the music and subtle subjective biases will tend to exert a strong influence over the work, due to the limits of the programmer's experience. Typically, the process of informal development is applied where musical problems are particularly obvious, with improvements accumulating rapidly to a prototype. Formal empirical methods are then applied to the prototype to detect more subtle problems. The first algorithm I developed for this research – the parametric morphing algorithm which is detailed in chapter five – relied only on my own personal evaluation. This is because I did not consider the algorithm to have reached a standard that was suitable for larger scale formal evaluation. Nevertheless, various musical examples and my own personal observations have been recorded, and these help to understand the affordances of the algorithm.

It should be noted that no formal comparison was made between the three algorithms developed through this project and other, historical algorithms. While this would have produced some interesting results, it is difficult to compare historical algorithms from differing musical contexts, and the continuous iterative development methodology did not call for comparisons between the three algorithms generated at each different stage of development.

Evaluating software

The software developed through this research is evaluated primarily in terms of whether or not it adequately demonstrates the musical algorithms I designed – this is because music is the primary focus, rather than software. Despite this, because the topic is *interactive* morphing, another important consideration is the time efficiency of the system, as expressed by 'Big O' notation. Usability of interface design is relevant insofar as I am able to demonstrate and perform using the software and thus no Human Computer Interface (HCI) evaluations were necessary. Extensibility of the software system and documentation is relevant on a personal level as this enables potential improvements in the future; however, it does not serve as a criterion for the success of the research.

1.7 Knowledge outcomes

The knowledge outcomes of this research are novel and significant. Note level morphing of MEM has been explored very little in the past, and there are a number of original aspects that have been investigated by this research, including new algorithmic music techniques, software system design and data gathering techniques. This is made clear towards the end of chapter three which provides a comprehensive review of the field of note level morphing. In terms of significance, there are a range of potential applications for note level morphing, as explained in 1.1 above and throughout the reviews in chapter two and three, including electronic music delivery, computer game music, accessible electronic music tools and computer assisted composition. This suggests that extended knowledge of such algorithms would be significant. As well as this, I have prepared some proof-of-concept demonstrations within some of these application contexts, as documented in chapter eight, the conclusion.

With this in mind, the four most significant and novel contributions made by this research will now be outlined. They are: a comprehensive review of note level morphing, a new software system for note level morphing algorithms, three new morphing algorithms and new data gathering techniques for examination of note level morphing algorithms.

1.7.1 Algorithms

I have developed three algorithms that fit the definition of source-target note level morphing: interpolation morph, Markov morph and *TraSe* morph.

In developing the interpolation morph, I took a parametric approach to note level morphing. It is similar to the algorithm described by Mathews and Rosler (1969), however it is more adaptive to changes that occur in realtime. Mathews and Rosler's algorithm is discussed in more detail in chapter three, while the interpolation morph that I developed is the topic of chapter five.

The Markov morph is a probabilistic approach to note level morphing. It provides realtime flexibility, is able to continuously generate new musical patterns and includes user-definable parameters that provide some degree of influence over the musicality of the output. It is distinguished from previous approaches to probabilistic note level morphing in that it utilises depths above one, that is, the conditional probability of preceding events, in conjunction with note similarity measures. The Markov morph is detailed in chapter six.

The *TraSe (Transform-Select)* morph utilises an evolutionary approach to note level morphing. It is successfully able to generate morphs that, for the most part, are discussed favourably in terms of smoothness and coherence. As well as this, *TraSe* circumvents the problem of requiring a data-driven approach – which would make it something other than "morphing" – by incorporating a range of compositional transformations to provide musical style. It is also very flexible: a number of different compositional transformation parameter values can be weighted so as to control the style of the resulting morph, the number of intervening states in the morph can be influenced and there are many other parameters, explained in more detail later on. The *TraSe* morph is particularly apt at performing a range of automatic key-modulations.

1.7.2 Data gathering techniques

Through this research I developed techniques for the formal examination of morphing algorithms as part of a focus concert and a web questionnaire. The most powerful technique was to ask the participants to record their subjective responses to important changes in the morph and to derive a reason for the response from the musical surface. This provided rich qualitative information into the various ways the music was being interpreted, highlighting potential improvements to the algorithm under examination as well as the subjective evaluation.

Importantly, the morphs generated by the morphing algorithm were benchmarked so as to enable the results to be viewed in a relevant context. For example, in the focus concert, the benchmark was a professional DJ who attempted to mix the source and target live. In the web questionnaire, a professional electronic music producer composed the morphs manually, using the same synthesis engine and MIDI files as used by *LEMorpheus*.

1.7.3 Contextual Review

The review of note level morphing systems presented in the third chapter and also the basis of a paper published at the peer-reviewed ‘Australasian Computer Music Conference 2006’ (Wooller 2006) provides a useful repository of diverse ideas and approaches to the problem. The review serves as a reference point for ascertaining the state and direction of note level morphing and for identifying topics in need of future development. It gathers together a range of formerly unconnected projects and defines note level morphing as a new area of research.

1.7.4 Software

LEMorpheus, the software application I developed for this project, provides a mechanism for the investigation of note level morphing algorithms. It has a modular design which allows the user to select between different morphing algorithms for experimentation and includes music representations and a system for rendering the morphs in realtime.

At the highest level, a simple interface provides control over which patterns and morphs are playing and the most important “morph index” parameter. The morph index can be controlled externally via MIDI. “Table mode” enables the morph index for each part to be controlled individually with *reacTIVision* fiducials (Jordà, Kaltenbrunner et al. 2005) through Open Sound Control (OSC, an alternative to MIDI).

The music representation affords control over key, scale and scale degree without losing the ability to represent passing notes. It is also extensible and may be applied to non-standard tuning systems. The supporting libraries provide a range of analytic, transformational and generative tools as well as algorithms that convert between various music representations.

1.8 Thesis structure

Having introduced the research motivations, goals, design and outcomes, more detailed explanations of various aspects of the research can be accessed in the subsequent chapters. An overview of the chapters will now be provided and may be used as a reference throughout the reading.

Chapter one introduces the goal, motivations, topic, research design, knowledge outcomes and thesis structure. The goal is to investigate techniques that both enable greater adaptivity and provide a new set of aesthetic possibilities for mainstream electronic music delivery. The motivation is in providing new capabilities in adaptive mainstream electronic music delivery contexts such as live electronic dance music, computer games, music for people with disabilities and others. The topic is note level morphing – the automatic generation of hybrid transitions between a source and target music. The research design is based on iterative software development and qualitative evaluation techniques. Particularly significant knowledge outcomes were the contextual review of morphing, the software system design, three new note level morphing algorithms and qualitative evaluation techniques specific to note level morphing.

Chapter two situates the thesis within the wider musical context. Firstly this involves a musicological definition of MEM and a review of current MEM practice that uses morphing or is morph-like. This is limited to composition or music production that is performed mostly through manual, rather than automated, processes. Following the discussion of MEM and morphing, the search for instances of manual note level morphing and similar practices is extended to genres of music outside of MEM.

Chapter three is also a contextual review, but with particular focus on algorithmic music systems rather than non-algorithmic music. The first two sections provide a terminological framework that is useful for comprehending and describing algorithmic music systems at the high and low-levels of encapsulation respectively. The framework is then applied to a series of three increasingly focused reviews, beginning with algorithmic composition, then to interactive music and finally note level morphing. The unique position of my own research within the field is made clear from this review.

Chapter four explains relevant aspects of the *LEMorpheus* software infrastructure that I developed to support experimentation with note level morphing algorithms. This includes explanations of: high level controls over morphing; the loop editor; parameters that affect the morph as a whole as well as each individual layer; music representations and extensible designs that support the note sequence morphing algorithms; and the system for rendering MIDI output in realtime. Some future developments for the software infrastructure are also summarised.

Chapter five details the parametric morphing algorithm, the first note level morphing algorithm I developed. It converts note data into separate continuous parameter envelopes which are then combined and weighted on the morph index during the morph and converted back to note data for playback. An overview of this process is given, followed by a more detailed, fully implementable description. Some informal evaluation and the audible results of the parametric morphing algorithm are then provided. The chapter finishes by detailing some extensions to the parametric morphing algorithm that could be implemented in the future.

Chapter six is concerned with the probabilistic morphing algorithm which I developed subsequently. This algorithm compares the most recent history with either source or target to create a matrix of similarity values, which is then used to predict the next note. Within chapter six an overview of this process provides some overall clarity. Following this, it is described to the level of detail necessary for implementation. The probabilistic morphing algorithm was evaluated both informally and formally though a qualitative "focus concert" study and the methods, results and analysis of these evaluations are also presented, along with audible examples. Problems and possible improvements to the current probabilistic morphing algorithm are then suggested.

Chapter seven explains the final, evolutionary morphing algorithm, *TraSe*. With *TraSe*, the source music is put through an iterative process of mutation and selection until the target music is arrived at. At each iteration, an array of mutants are created by a set of compositional transformations, one of which is selected based on a measure of similarity to the target. After the overview and detailed discussion of *TraSe*, the informal and formal evaluation of it is presented, along with audible examples. The formal evaluation was conducted through a qualitative online survey and the methods, results and analysis of which are included. Following this, possible improvements to *TraSe* are suggested and an optimisation technique that would allow greater realtime interactivity is scoped out in detail.

Chapter eight concludes the thesis, beginning with some demonstration examples of potential applications of note level morphing that I have prototyped, namely, computer games and live collaborative music making. Some future research possibilities that would either extend and/or complement the current study are then discussed. Finally, some concluding remarks are expressed, relating the contributions that have been demonstrated through the thesis back to the primary research objectives.

With the structure of the thesis summarised for easy reference, the research can be explained in detail. As the research is fundamentally an exploration of musical processes, the logical starting point for this explanation is the surrounding musical context, the topic of the following chapter.