COMPER: TOWARDS A MODEL FOR GENERATING COMPOSITIONS FROM EXPRESSIVE MUSIC PERFORMANCES

Alexis Kirke
Interdisciplinary Centre for Computer Music Research, School of Computing, Communications and Electronics, University of Plymouth, Plymouth, UK

Eduardo R. Miranda
Interdisciplinary Centre for Computer Music Research, School of Computing, Communications and Electronics, University of Plymouth, Plymouth, UK

ABSTRACT

We introduce the concept of Expressive Performance Generated Composition where the initial specification for a composition comes from the details of an expressive performance. This expressive performance may have been of a previous piece or be specially produced for the composition. The expressive deviations in the performance are used to generate a meaningful interpretation of the performance through a new composition, by attempting to invert methodologies for simulating human expressive performance of music. The development of such systems is motivated by four elements: (1) The final composed piece will be immediately playable by computer in an expressive way without any need for complex structural analysis; (2) It provides methods for an innovative, but meaningfully specified, system for composition; (3) Research done into computer expressive performance can then be tested for application to algorithmic composition – for example emotional expression; (4) It may be easier for inexperienced composers/musicians to specify a composition by expressive performance elements.

1. INTRODUCTION

A computer will perform most MIDI files in perfect metronomic time, which usually sounds inhuman. This is because human performers adjust the microstructure of the music – they perform a score expressively. For example: speeding up and slowing down while playing, and changing how loud they play. The performer's additions allow them to express a fixed score, hence the term "Expressive" Performance. The last 25 years has seen significant amounts of research into Computer System for Expressive Music Performance [1].

However this research has generally been done in isolation to research into algorithmic composition. But in reality composition and expressive performance are not such separate phenomena. [2] observes that "the separation of musical rules into structural and performative is largely an ontological one, and cedes nothing to the final audio experienced by the listener." The dichotomy is often just between macrofeatures and microfeatures of the music. However the composition process can include the generation of microfeatures; for example some compositions by members of the New Complexity school [3]. So providing tools for generating microfeatures is not only desirable in CSEMPs, but also in computer (or computer-aided) composition.

There is another, less artistic, reason for bringing computer composition and expressive performance closer. Previous research into human performance has shown that the analytical structure of the score has a large impact on the expressive performance additions [4]. So a significant amount of computational/research effort in systems for expressive performance is in analyzing the musical structure of expressionless input. However, many computer composition systems generate a piece based on some structure which can often be made explicitly available by the system. So in computer music it is often inefficient to have separate composition and expressive performance systems – where a score is generated and CSEMP sees the score as a black box and performs a structure analysis. Greater efficiency and accuracy would require a protocol allowing the computer composition system to communicate structure information directly to the computer system for expressive performance, or simply combine the systems using for example a common representation (where microtiming and microdynamics are seen as an actual part of the composition process). Such approaches are also advantageous for algorithmic composition systems that generate less traditional music which is to be played by a computer. Most computer systems for expressive performance are based on ideas of score analysis which are more relevant to pre-1940s music – there are far fewer musicological tools available for analyzing the structure of post-1940s music [5].

2. A BRIEF REVIEW OF COMBINED COMPOSITION AND EXPRESSIVE PERFORMANCE SYSTEMS

There has been some research into combined composition and performance systems and we will look at 3 examples. The Computational Music Emotion Rule System (CMERS) [2] has a rule-set of 19 rules developed through analysis-by-synthesis. These rules are designed not only to inject microfeature deviations into the score to generate human-like performances, but also to use microfeature and macrofeature deviations to express emotions to the listener. To this end CMERS is able to change the score itself, recomposing it. CMERS has a 2-D model of human emotion space to give such emotions as angry, bright, contented and despairing. The rules for expressing emotions include: moving between major and minor modes, changing note pitch classes, as well as rules for small changes in dynamics and tempo. A significant number of formal listening tests have been done and examples of CMERS are available on the author's webpage.

Ossia [6] is a system which incorporates both compositional and performance aspects. However, whereas CMERS was designed to operate on a composition, Ossia is able to generate entirely new and expressively performed compositions. Ossia generates music through a novel representational structure that encompasses both composition and performance – Recursive Trees (generated by Genetic Algorithms). These are "upside down trees" containing both performance and composition information. The bottom leaves of the tree going from left to right represent actual notes (each with their own pitch, duration and loudness value) in the order they are played. The branches above the notes represent transformations on those notes. To generate music the tree is flattened – the "leaves" higher up act upon the leaves lower down when being flattened to produce a performance/composition. So going from left to right in the tree represents music in time. The trees are generated recursively – this means that the lower branches of the tree are transformed copies of higher parts of the tree. Here we have an element we argue is the key to combined performance and composition systems - a *common representation* – in this case Transformations.

Most computer expressive performance systems transform an expressionless MIDI or audio file into an expressive version [1]. Composition is often done in a similar way – motifs are transformed into new motifs, and themes are transformed into new expositions. Ossia uses a novel transformation-based music representation. In Ossia, transformations of note, loudness and duration are possible – the inclusion of note transformations here emphasising the composition aspect of the Ossia. The embedding of these transformations into recursive trees leads to the generation of gradual crescendos, decrescendos and duration curves – which sound like performance strategies to a listener. Because of this Ossia has a good level of performance creativity. The trees also create a structure of themes and expositions. Ossia uses a Genetic Algorithm to generate a population of trees, and judges for fitness using such rules as number of notes per second, repetivity, amount of silence, pitch variation, and level of recursion. These fitness rules were developed heuristically by Dahlstedt through analysis-by-synthesis methods. Examples are also available on Dahlstedt's website, including a composed suite.

The final system we will review at is a Multi-agent System [7] called pMIMACS [8]. The aim of pMIMACS is to generate contemporary compositions on a computer which when played back on a computer do not sound too machine like. The agent cycle is a process of singing and assimilation. Initially all agents are given their own tune – these may be random or chosen by the composer. An agent (A) is chosen to start. Agent A performs its tune, based on its "performance skill" (explained below). All other agents listen to A and the agent with the most similar tune, say agent B, adds its interpretation of A's tune to the start or end of its current tune. There may be pitch and timing errors due to its "mishearing". Then the cycle begins again, but with B performing its extended tune in the place of A.

An agent's initial performing skills are defined by the average pitch and standard deviation of their initial tune - this could be interpreted as the tune they are familiar with performing, or as the range they are comfortable performing in. The further away a note's pitch is from the agent's average learned pitch, the slower the tempo at which the agent will perform it. Also, further away pitches will be played more quietly. An agent updates its skill/range as it plays. Every time it plays a note, that note changes the agent's average and standard deviation pitch value. So when an agent adds an interpretation of another agent's tune to its own, then as the agent performs the new extended tune its average and standard deviation (skill/range) will update accordingly – shifting and perhaps widening - changed by the new notes as it plays them. In pMIMACS an agent also has an Excitability State. An "excited" agent will play its tune with twice the tempo of an "unexcited" agent, making macro-errors in pitch and rhythm as a result.

The listening agent has no way of knowing whether the pitches, timings and amplitude that it is hearing are due to the performance skills of the performing agent, or part of the "original" composition. So the listening agent attempts to memorize the tune as it hears it, including any performance errors or changes. As the agents perform to each other, they store internally and exponentially growing piece of transforming music. The significant and often smooth deviations in tempo that pMIMACS generates are suggestive of human expressive performances, and hence create a more pleasant sounding performance than if the rhythms had been generated using a simple rhythm palette with no consideration of a flow of tempo microstructure. Examples of an agent's tune memory after a number of cycles can be listened to on the authors' website.

3. GENERATING COMPOSITIONS FROM EXPRESSIVE PERFORMANCES

In this paper we introduce a new methodology for bringing algorithmic composition and expressive performance closer together: constraint-based composition using expressive performance descriptions for constraints. Constraint-based systems have been widely researched and are in practical use in algorithmic composition [9]. A typical expressive performance constraint might be: "accelerando starting at score point A until score point B"; or "crescendo from point D until point E".

Taking into account our comments in the introduction, such an approach has the following advantages:

1) The final composed piece will be immediately playable by computer in an expressive way without any need for complex structural analysis.
2) It provides methods for an innovative, but meaningfully specified, system for composition.
3) Research done into computer expressive performance can then be tested for application to algorithmic composition – for example emotional expression [10].
4) It may be easier for inexperienced composers/musicians to specify a composition by expressive performance elements.

We have already covered point (1). Looking first at Point (3) - there has been much research into expressive performance which could now be utilized by algorithmic composition. For example there have been a number of systems that link expressive performance actions to the emotional experience of the listener, for example [2] and [10]. Thus there is the potential to use such systems to generate composition with a certain emotional content. Point (4) refers to the idea those without musical training may find it more intuitive to generate a piece of music through loudness and speed, than through the use of keys, pitches and note-timing values. This would open up composition to a larger number of people, including children in musical education. Furthermore dynamics and tempo may be more naturally linked to gestural control – so one could imagine a number of interfaces – motion-sensor batons, "air guitars", etc which could be used to generate music by non-musicians. And if such music was generated using expressive performance constraints it may be more likely to fulfill the amateur composer's expectations. Point (2) simply refers to the fact that such a novel approach to composition could provide creative opportunities even to more experienced composers.

4. COMPER – A PERFORMANCE-CONSTRAINED COMPOSITION SYSTEM

As an example of utilizing the ideas in section 3, we introduce our composition system COMPER (Composition by expressive Performance). COMPER is a simple alleatoric algorithmic composition system that uses as constraints a subset of well-known performance rules. These rules are taken from the Director Musices system for music performance. We refer the reader to [11] for a detailed explanation of them:

a) *Phrase Arch* – The user can set phrase lengths by defining a series of expressive accelerando's and deccelerando's. If an accelerando is followed by a deccelerando, then this rule will define the start of the accelerando as the start of a note grouping, and the end of the deccelerando as the end of a note grouping.

b) *High Loud* – The louder the user defines a part of the piece to be, the greater its tendency to be a high pitch.

c) *Melodic Charge* – The louder a note is defined, the more likely it is that a "highly charged" pitch will be selected.

d) *Faster upward run* – The faster the user defines the section as being, the more likely notes will go upwards.

e) *Chromatic Emphasis* – The louder a user defines the section to be, the more semi-tone intervals will be used.

Apart from phrase start and end definitions, these constraints are stochastic in COMPER – they just make certain pitches and rhythms more likely. The relative effect of each constraint is defined by what is known in Director Musices as the "k-value". A value of 0 will mean the constraint is not applied at all, and a higher k-value causes the constraint to be applied with greater frequency (a negative k-value causes the constraint to be reversed).

Most of the constraints are self-explanatory if only partially defined here (due to space reasons). But one of the less obvious constraints is the phrase arch rule, which we will now define more clearly. Based mainly on this rule, COMPER defines the length of the motifs that it generates. To define the phrase arch constraints, the user sets up a series of expressive accelerandos and deccelerandos. Suppose there are 8 expressive accelerandos with start points: AS1 up to AS8; and the end points of the interspersed deccelerandos are DE1 to DE8. These define a set of phrases going from AS1->DE1, AS2->DE2, etc. Now we will have a set of phrase lengths defined (DE1–AS1) up to (DE8-AS8). COMPER will look for any common phrase lengths. Suppose all of the phrases are either of length 8 beats or length 24 beats. Then COMPER will only generate motifs of length 8 and 24, and will be constrained to place these motifs as defined by phrase starts and end. For example three of the phrase lengths could be 24 beats and 4 of them 8 beats. To create a more interesting composition there should generally be less motifs generated than there are phrase spaces – to allow for repeats. This rule could be extended into a hierarchical version, where the user defines the start and end of themes and movements, as well as motifs – thus allowing a more complex structure to be constrained.

So to constrain the piece based on the above rules, the user can then specify:

1. A series of tempo values
2. A series of dynamics values

and based on these COMPER can generate an alleatoric composition. For a given set of phrase starts and settings for (1) and (2), multiple compositions are possible based on the COMPER constraints. As has already been mentioned – the above parameters could be specified in a number of suggestive ways – e.g. a baton, or a suitable GUI and touch-screen. Because of space restrictions we will not report any experimental results here, but leave this for a more detailed report.

One element that needs to be addressed is that there is an advantage and disadvantage to using the Director Musices rules. The advantage is they are easy to understand and relatively simple. The disadvantage is that when inverted they are such broad generalisations that they could have a tendency to generate pieces which are too similar (e.g. should loud parts *always* tend to high pitches, should faster parts *always* tend to run upwards?) This is not necessarily a failing of the DM rules (which are designed for performance), but of their application to the compositional domain. There are computer expressive performance systems with more sophisticated rule sets, which could be used to generate more sophisticated compositions. However, we present COMPER as a simple first step in the process of expressive performance based compositions.

5. CONCLUSIONS

In this paper we have introduced a new methodology for bringing algorithmic composition and expressive performance closer together: constraint-based composition using expressive performance descriptions for constraints. We have discussed motivations for bringing computer composition and performance closer, and in particular motivations for using expressive performance to generate compositions. As a first step in this direction we have introduced the alleatoric composition system COMPER which bases its constraints on the rules of Director Musices, the well-known expressive performance system for computers.

In terms of future work, we would like to investigate more sophisticated ways of turning expressive performance rules into constraints, and also more varied sets of rules for expressive performance. We hope that our example and discussion here will encourage more research into this fruitful area.

This work has been funded by the UK EPSRC project “Learning the Structure of Music” (EPD063612-1).

REFERENCES

[1] Kirke, A. and Miranda, E. A Survey of Computer Systems for Expressive Music Performance, *ACM Computing Surveys*, In Print, 2008

[2] Livingstone, S.R., Muhlberger, R., Brown, A.R., and Loch, A. Controlling Musical Emotionality: An Affective Computational Architecture for Influencing Musical Emotions. *Digital Creativity* 18, Taylor and Francis 2007

[3] Toop, R. Four Facets of the New Complexity. *CONTACT* 32, 4-50, 1988

[4] Palmer, C. Music performance. *Annual Review of Psychology* 48, 115-138, 1997

[5] Clarke, E.F. Expression and communication in musical performance. In Sundberg, J., Nord, L., and Carlson, R., Eds. Macmillan Press, London, 1991

[6] Dahlstedt, P.. Autonomous Evolution of Complete Piano Pieces and Performances. In *Proceedings of ECAL 2007 Workshop on Music and Artificial Life* (MusicAL 2007), Lisbon, Portugal, September 2007.

[7] Kirke, A., *Learning and Co-operation in Multi-Robot Systems.* PhD Thesis, University of Plymouth 1997

[8] Kirke, A., Miranda, E.R., Using a Biophysically-constrained Multi-agent System to combine Expressive Performance with Algorithmic Composition, In Print, 2008

[9] Anders, T.. *Composing Music by Composing Rules: Design and Usage of a Generic Music Constraint System.* PhD Thesis, University of Belfast, 2007

[10] Bresin, R. and Friberg, A. Emotional Coloring of Computer-Controlled Music Performances. *Computer Music Journal* 24, 44-63, 2000

[11] Friberg, A., Bresin, R., and Sundberg, J. Overview of the KTH rule system for musical performance. *Advances in Cognitive Psychology* 2, 145-161, Vizja Press & IT Ltd 2006