An Overview of Computer Systems for Expressive Music Performance

Alexis Kirke and Eduardo R. Miranda

Abstract

This chapter is a survey of research into automated and semi-automated computer systems for expressive performance of music. We examine the motivation for such systems and then examine a significant sample of the systems developed over the last 30 years. To highlight some of the possible future directions for new research, this chapter uses primary terms of reference based on four elements: testing status, expressive representation, polyphonic ability and performance creativity.

1.1 Introduction

Computer composition of classical music had been around since 1957 when the Illiac Suite for String Quartet—the first published composition by a computer—was published by Lejaren Hiller [1]. Since then there has been a large body of such music and research published, with many successful systems produced for automated and semi-automated computer composition [2–4]. But publications on computer expressive performance of music lagged behind composition by almost quarter of a century. During the period when MIDI and computer use exploded amongst pop performers and up to 1987—when Yamaha had released their first Disklavier MIDI piano—there were only two or three researchers publishing on algorithms for expressive performance of music [5] including the KTH system discussed in Chaps. 2 and 7. However, from the end of the 1980s onwards, there

A. Kirke (✉) • E.R. Miranda
Interdisciplinary Centre for Computer Music Research, Plymouth University,
Faculty of Arts, Drake Circus, PL4 8AA Plymouth, UK
e-mail: alexis.kirke@plymouth.ac.uk; eduardo.miranda@plymouth.ac.uk

was an increasing interest in automated and semi-automated computer systems for expressive music performance (CSEMP). A CSEMP is a computer system able to generate expressive performances of music. For example, software for music typesetting will often be used to write a piece of music, but some packages play back the music in a relatively robotic way—the addition of a CSEMP enables a more realistic playback. Or an MP3 player could include a CSEMP which would allow performances of music to be adjusted to different performance styles.

This book contains a description of various systems and issues found in CSEMP work. As an introduction, this chapter provides an overview of a significant sample of research on automated and semi-automated CSEMPs. By automated, we refer to the ability of the system—once set up or trained—to generate a performance of a new piece, not seen before by the system, without manual intervention. Some automated systems may require manual set-up but then can be presented with multiple pieces which will be played autonomously. A semi-automated system is one which requires some manual input from the user (e.g. a musicological analysis) to deal with a new piece.

1.1.1 Human Expressive Performance

How do humans make their performances sound so different to the so-called perfect performance a computer would give? In this chapter, the strategies and changes which are not marked in a score but which performers apply to the music will be referred to as expressive performance actions. Two of the most common performance actions are changing the tempo and the loudness of the piece as it is played. These should not be confused with the tempo or loudness changes marked in the score, like accelerando or mezzo forte, but to additional tempo and loudness changes not marked in the score. For example, a common expressive performance strategy is for the performer to slow down as they approach the end of the piece [6]. Another performance action is the use of expressive articulation—when a performer chooses to play notes in a more staccato (short and pronounced) or legato (smooth) way. Those playing instruments with continuous tuning, for example, string players, may also use expressive intonation, making notes slightly sharper or flatter, and such instruments also allow for expressive vibrato. Many instruments provide the ability to expressively change timbre as well.

Why do humans add these expressive performance actions when playing music? We will set the context for answering this question using a historical perspective. Pianist and musicologist Ian Pace offers up the following as a familiar historical model for the development of notation (though suggests that overall it constitutes an oversimplification) [7]:

> In the Middle Ages and to a lesser extent to the Renaissance, musical scores provided only a bare outline of the music, with much to be filled in by the performer or performers, freely improvising within conventions which were essentially communicated verbally within a region or locality. By the Baroque Era, composers began to be more specific in terms of requirements for pitch, rhythm and articulation, though it was still common for performers

to apply embellishments and diminutions to the notated scores, and during the Classical Period a greater range of specificity was introduced for dynamics and accentuation. All of this reflected a gradual increase in the internationalism of music, with composers and performers travelling more widely and thus rendering the necessity for greater notational clarity as knowledge of local performance conventions could no longer be taken for granted. From Beethoven onwards, the composer took on a new role, less a servant composing to occasion at the behest of his or her feudal masters, more a freelance entrepreneur who followed his own desires, wishes and convictions, and wrote for posterity, hence bequeathing the notion of the master-work which had a more palpable autonomous existence over and above its various manifestations in performance. This required an even greater degree of notational exactitude; for example in the realms of tempo, where generic Italianate conventions were both rendered in the composer's native language and finely nuanced by qualifying clauses and adjectives. Through the course of the nineteenth century, tempo modifications were also entered more frequently into scores, and with the advent of a greater emphasis on timbre, scores gradually became more specific in terms of the indication of instrumentation. Performers phased out the processes of embellishment and ornamentation as the score came to attain more of the status of a sacred object. In the twentieth century, this process was extended much further, with the finest nuances of inflection, rubato, rhythmic modification coming to be indicated in the score. By the time of the music of Brian Ferneyhough, to take the most extreme example, all minutest details of every parameter are etched into the score, and the performer's task is simply to try and execute these as precisely as he or she can.

So in pre-twentieth-century music there has been a tradition of performers making additions to a performance which were not marked in the score (though the reason Pace calls this history an oversimplification is that modern music does have the capacity for expressive performance, as we will discuss later).

A number of studies have been done into this pre-twentieth-century (specifically baroque, classical and romantic) music performance. The earliest studies began with Seashore [8], and good overviews include Palmer [9] and Gabrielsson [10]. One element of these studies has been to discover what aspects of a piece of music—what musical features—are related to a performer's use of expressive performance actions. One of these musical features expressed is the performer's structural interpretation of the piece [9]. A piece of music has a number of levels of meaning—a hierarchy. Notes make up motifs, motifs make up phrases, phrases make up sections and sections make up a piece (in more continuous instruments, there are intranote elements as well). Each element—note, motif, etc.—plays a role in other higher elements. (This is discussed in more depth in Chap. 8.) Human performers have been shown to express this hierarchical structure in their performances. Performers have a tendency to slow down at boundaries in the hierarchy—with the amount of slowing being correlated to the importance of the boundary [11]. Thus, a performer would tend to slow more at a boundary between sections than between phrases. There are also regularities relating to other musical features in performers' expressive strategies. For example, in some cases the musical feature of higher-pitched notes causes a performance action of the notes being played more loudly; also, notes which introduce melodic tension relative to the key may be played more loudly. However, for every rule there will always be exceptions.

Another factor influencing expressive performance actions is performance context. Performers may wish to express a certain mood or emotion (e.g. sadness, happiness)

through a piece of music. Performers have been shown to change the tempo and dynamics of a piece when asked to express an emotion as they play it [12]. For a discussion of other factors involved in human expressive performance, we refer the reader to [13].

1.1.2 Computer Expressive Performance

Having examined human expressive performance, the question now becomes: why should we want computers to perform music expressively? There are at least five answers to this question:

1. *Investigating human expressive performance by developing computational models*—Expressive performance is a fertile area for investigating musicology and human psychology [8–10]. As an alternative to experimentation with human performers, models can be built which attempt to simulate elements of human expressive performance. As in all mathematical and computational modelling, the model itself can give the researcher greater insight into the mechanisms inherent in that which is being modelled.
2. *Realistic playback on a music typesetting or composing tool*—There are many computer tools available now for music typesetting and for composing. If these tools play back the compositions with expression on the computer, the composer will have a better idea of what their final piece will sound like. For example, Sibelius, Notion and Finale have some ability for expressive playback.
3. *Playing computer-generated music expressively*—There are a number of algorithmic composition systems that output music without expressive performance but which audiences would normally expect to hear played expressively. These compositions in their raw form will play on a computer in a robotic way. A CSEMP would allow the output of an algorithmic composition system to be played directly on the computer which composed it (e.g. in a computer game which generates mood music based on what is happening in the game).
4. *Playing data files*—A large number of non-expressive data files in formats like MIDI and MusicXML [14] are available on the Internet, and they are used by many musicians as a standard communication tool for ideas and pieces. Without CSEMPs, most of these files will play back on a computer in an unattractive way, whereas the use of a CSEMP would make such files much more useful.
5. *Computer accompaniment tasks*—It can be costly for a musician to play in ensemble. Musicians can practise by playing along to recordings with their solo part stripped out. But some may find it too restrictive since such recordings cannot dynamically follow the expressiveness in the soloist's performance. These soloists may prefer to play along with an interactive accompaniment system that not only tracks their expression but also generates its own expression.

1.2 A Generic Framework for Previous Research in Computer Expressive Performance

Figure 1.1 shows a generic model for the framework that many (but not all) previous automated and semi-automated CSEMPs tend to have followed. The modules of this diagram are described here.

Performance Knowledge—This is the core of any performance system. It is the set of rules or associations that control the performance action. It is the 'expertise' of the system which contains the ability, implicit or explicit, to generate an expressive performance. This may be in the form of an artificial neural network, a set of cases in a case-based reasoning system or a set of linear equation with coefficients. To produce performance actions, this module uses its programmed knowledge together with any inputs concerning the particular performance. Its main input is the music/analysis module. Its output is a representation of the performance of the musical input, including expressive performance actions.

Music/Analysis—The music/analysis module has two functions. First of all, in all systems, it has the function of inputting the music to be played expressively (whether in paper score, MIDI, MusicXML, audio or other form) into the system. The input process can be quite complex, for example, paper score or audio input will require some form of analytical recognition of musical events. This module is the only input to the performance knowledge module that defines the particular piece of music to be played. In some systems, it also has a second function—to provide an analysis of the musical structure. This analysis provides information about the music features of the music—for example, metrical, melodic or harmonic structure. (It was mentioned earlier how it has been shown that such structures have a large influence on expressive performance in humans.) This analysis can then be used by the performance knowledge system to decide how the piece should be performed. Analysis methods used in some of the systems include Lerdahl and Jackendoff's generative theory of tonal music discussed in Chap. 8 [15], Narmour's implication realisation discussed in Chap. 3 [16] and various bespoke musical measurements. The analysis may be automated, manual or a combination of the two.

Performance Context—Another element which will affect how a piece of music is played is the performance context. This includes such things as how the performer decides to play a piece, for example, happy, perky, sad or lovelorn. It can also include whether the piece is played in a particular style, for example, baroque or romantic.

Adaptation Process—The adaptation process is the method used to develop the performance knowledge. Like the analysis module, this can be automated, manual or a combination of the two. In some systems, a human expert listens to actual musical output of the performance system and decides if it is appropriate. If not, then the performance knowledge can be adjusted to try to improve the musical output performance. This is the reason that in Fig. 1.1 there is a line going from the sound module back to the adaptation procedure module. The adaptation procedure
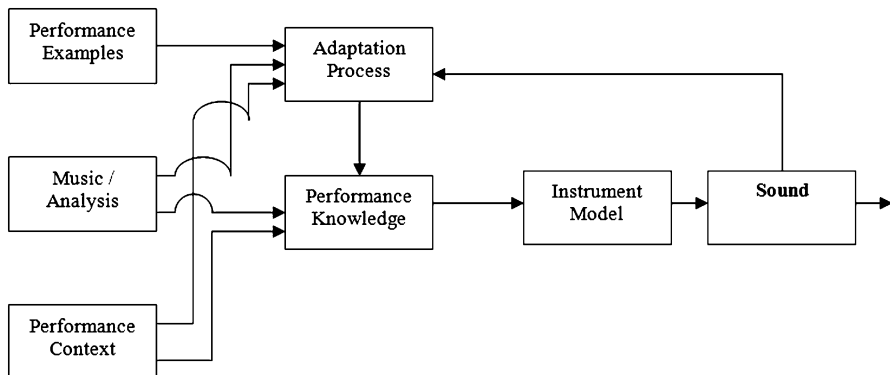
Fig. 1.1 Generic model for most current CSEMPs

also has inputs from performance context, music/analysis, instrument model and performance examples. All four of these elements can influence the way that a human performs a piece of music, though the most commonly used are music/analysis and performance examples.

Performance Examples—One important element that can be incorporated in the performance knowledge building is the experience of past human performances. These examples can be used by the adaptation procedure to analyse when and how performance actions are added to a piece of music by human performers. The examples may be a database of marked-up audio recordings, MIDI files together with their source scores or (in the manual case) a person's experience of music performance.

Instrument Model—By far the most common instrument used in computer-generated performance research is the piano. This is because it allows experiments with many aspects of expression but requires only a very simple instrument model. In fact, the instrument model used for piano is often just the MIDI/media player and sound card in a PC. Alternatively, it may be something more complex but still not part of the simulation system, for example, a Yamaha Disklavier (e.g. as used in Chap. 7). However, a few simulation systems use non-keyboard instruments, for example, violin (used in Chap. 5) and trumpet. In these cases, the issue of a performance is more than just expressiveness. Just simulating a human-like performance, even if it is non-expressive, on these instruments is non-trivial. So systems simulating expressive performance on such instruments may require a relatively complex instrument model in addition to expressive performance elements. (Chaps. 5 and 9 use more complex instrument models.)

1.2.1   Modules of Systems Reviewed and Terms of Reference

Table 1.1 lists the systems reviewed, together with information about their modules. This information will be explained in the detailed part of the chapter. Note that

Table 1.1 Systems reviewed

CSEMP	Performance knowledge	Music	Analysis	Performance context	Adaptation process	Performance examples	Instrument model (or application)	Performance actions
Director Musices [17] Chap. 2/7	Rules	MIDI, score	Custom	Mood space	–	MIDI performances	All (piano)	T/D/A/P
Hierarchical parabola [18–20]	Parabola equation	MIDI, score	GTTM TSR	–	–	–	Piano	T/D
Composer pulse [21, 22]	Multiplier set	MIDI	–	–	Manual	Tapping	All	T/D
Bach fugue [23]	Rules	Score	Custom	–	Manual	Books, experts	Keyboard	T/A
Trumpet synthesis [24, 25]	Linear model	Audio, score	Custom	–	Manual	Audio performances	Trumpet	A/P/O
Rubato [26, 27]	Operators	MIDI, score	Custom	–	Manual	–	All (piano)	T/D
Pop-E [28]	Rule-based	MIDI, score	GTTM, custom	–	–	MIDI performances	Piano	T/D
Hermode tuning [29]	Rule-based	MIDI	Custom	–	–	–	All	P
Sibelius [30]	Rule-based	Score	Custom	–	–	Unknown	–	Unknown
CMERS [31]	Linear model	MIDI, score	Custom	Mood space	–	Performances and experiments	Piano	T/D/P/O
MIS [32–34]	Linear model	MIDI, score	GTTM/Meyer	–	Regression with ANDs	Audio performances	Piano	T/D/A
CaRo [35–37]	Linear model	Audio	Custom	Mood space	PCA, linear regression	Audio performances by mood	All	T/K/D/A
ANN piano [38]	ANN	MIDI	Custom	–	ANN training	MIDI performances	Piano	T/D
Emotional flute [39]	ANN and rules	Audio, score	Custom	Mood space	ANN training	Audio performances by mood	Flute	T/D/V

(continued)

Table 1.1 (continued)

CSEMP	Performance knowledge	Music	Analysis	Performance context	Adaptation process	Performance examples	Instrument model (or application)	Performance actions
SaxEx [40–42]	Fuzzy rules	Audio, score	Narmour/IR/GTTM, custom	Mood space	CBR training	Audio performances by mood	Saxophone	T/D/V/K
Kagurame [43, 44]	Rules	MIDI, score	Custom	Performance conditions	CBR training	MIDI performances by context	Piano	T/D/A
Ha-Hi-Hun Chap. 8	Rules	MIDI, score	GTTM TSR	Language performance conditions	CBR training	MIDI performances by condition	Piano	T/D
PLCG [46–48]	Learned rules	MIDI, score	Custom	–	Meta-sequential learning	MIDI performances	Piano	T/D/A
PLCG/Phrase-decomposition [49]	Learned rules	MIDI, score	Custom, harmonic by musicologist	–	CBR training, meta-sequential learning	MIDI performances	Piano	T/D/A
DISTALL Chap. 3 [50, 51]	CBR	MIDI, score	Custom, harmonic by musicologist	–	CBR training	MIDI performances	Piano	T/D/A
Music Plus One [52–54]	BBN	Audio/MIDI, score	Custom	Soloist tempo	BBN training	Audio soloist performances	All	T/D
ESP piano [55]	HMM	MIDI, score	Custom	–	HMM training	MIDI performances	Piano	T/D/A
Drumming [56]	Non-linear mapping	Audio	Custom	–	KRR, GPR, kNN	Audio performances	Drums	T
KCCA piano [57]	Non-linear mapping	Worm, score	Custom	–	KCCA	Performance worm	Piano	T/D

Genetic programming Chap. 5 [58]	Regression trees	Audio	IR, custom	–	Genetic programming	Audio performances	Saxophone	T/D/A/N
Generative performance GAs Chap. 4 [59]	Pulse set	MIDI, score	LBDM, Kruhmansl, Melisma	–	GA	None	Piano	T/D
MAS with imitation Chap. 4 [60]	Pulse set	MIDI, score	LBDM, Kruhmansl, Melisma	–	Imitation	None	Piano	T/D
Ossia [61]	Fitness rules	MIDI	–	–	–	None	Piano	T/D/P
pMIMACS	Rule-based (2)	MIDI	Performance skill	Excitability state	Imitation	None	Piano	T/D

Table 1.2 Abbreviations

Abbreviation	Meaning
A	Articulation
ANN	Artificial neural network
BBN	Bayesian belief network
CBR	Case-based reasoning
CSEMP	Computer system for expressive music performance
D	Dynamics
DM	Director musices (KTH system)
EC	Evolutionary computing
GA	Genetic algorithm
GP	Genetic programming
GPR	Gaussian process regression
GTTM	Lerdahl and Jackendoff's generative theory of tonal music
GUI	Graphical user interface
HMM	Hidden Markov model
IBL	Instance-based learning
IR	Narmour's implication/realisation theory of melody
K	Attack
KCCA	Kernel canonical correlation analysis
kNN	k-Nearest neighbour
KRR	Kernel ridge regression
LBDM	Local boundary detection model of Cambouropoulos
MAS	Multi-agent system
CMERS	Computational music emotion rule system by Livingstone et al.
MIDI	Musical instrument digital interface
MIMACS	Mimetics-inspired multi-agent composition system
MusicXML	Music extensible markup language
MIS	Music interpretation system by Katayose et al.
N	Note addition/consolidation
P	Pitch
PCA	Principal component analysis
T	Tempo
TSR	Time span reduction technique (from GTTM)
V	Vibrato

the column for instrument model is also used for CSEMPs without an explicit instrument model so as to show their applicability to various instruments. A number of abbreviations are used in Table 1.1 and throughout the chapter. Table 1.2 lists these abbreviations and their meaning.

Before discussing the primary terms of reference for this chapter, it should be observed that the issue of evaluation of CSEMPs is an open problem. How does one evaluate what is essentially a subjective process? If the CSEMP is trying to simulate a particular performance, then correlation tests can be done. However, even if the correlations are low for a generated performance, it is logically possible for the generated performance to be more preferable to some people than the

original performance. Chapter 7 goes into detail about performance evaluation in CSEMPs, but we will briefly address it here. Papadopoulos and Wiggins [62] discuss the evaluation issue in a different but closely related area—computer algorithmic composition systems. They list four points that they see as problematic in relation to such composition systems:

1. The lack of evaluation by experts, for example, professional musicians.
2. Evaluation is a relatively small part of the research with respect to the length of the research paper.
3. Many systems only generate melodies. How do we evaluate the music without a harmonic context? Most melodies will sound acceptable in some context or other.
4. Most of the systems deal with computer composition as a problem solving task rather than as a creative and meaningful process.

All of these four points are issues in the context of computer systems for expressive performance as well. So from these observations, three of the primary terms of reference are extracted for this chapter. The three primary terms of reference which are extracted from points 1–4 are performance testing status (points 1 and 2), polyphonic ability (point 3) and performance creativity (point 4). These first three dimensions will now be examined, and then the fourth primary term of reference will be introduced.

1.2.2 Primary Terms of Reference for Systems Surveyed

Performance testing status refers to how and to what extent the system has been tested. It is important to emphasise that testing status is not a measure of how successful the testing was, but how extensive it was. (This is also discussed in Chap. 7.) There are three main approaches to CSEMP testing: (a) trying to simulate a particular human performance or an average of human performances, (b) trying to create a performance which does not sound machine-like and (c) trying to create as aesthetically pleasing a performance as possible. For the first of these, a correlation can be done between the computer performance and the desired target performance/performances. (However, this will not be a 'perceptually weighted' correlation; errors may have a greater aesthetic/perceptual effect at some points than at others.) For approaches (b) and (c), we have listening tests by experts and non-experts. A wide variety of listening tests are used in CSEMPs, from the totally informal and hardly reported to the results of formal competitions.

Each year since 2002, a formal competition that has been described as a 'musical turing test', called the RenCon (contest for performance rendering systems) Workshop, has been held [63]. This will be discussed in detail in Chap. 7. About a third of the systems we will survey have been entered into a RenCon competition (see Table 1.3 for the results). RenCon is a primarily piano-based competition for baroque, classical and romantic music and includes manual as well as automated systems (though the placings in Table 1.3 are displayed relative to automated and semi-automated CSEMPs, ignoring the manual submissions to RenCon). Performances are graded and voted on by a jury of attendees from the sponsoring

Table 1.3 RenCon placings of CSEMPs in this chapter

CSEMP	RenCon placings
Director Musices	2002 (4th), 2004 compulsory (1st), 2005 compulsory (2nd)
SuperConductor (includes composer pulse)	2004 open (1st), 2006 open (1st and 4th)
Rubato	2004 open (4th)
MIS	2002 (2nd)
Music Plus One	2003 (1st)
Kagurame	2002 (6th), 2004 compulsory (3rd), 2006 compulsory (2nd)
Ha-Hi-Hun	2002 (5th), 2004 compulsory (4th), 2006 compulsory (3rd)
DISTALL	2002 (1st)
Pop-E	2005 compulsory (1st), 2006 compulsory (1st), 2006 open (2nd and 3rd)

conference. Scores are given for 'humanness' and 'expressiveness', giving an overall 'preference' score. It is the preference score we will focus on in the survey. The size of RenCon juries and their criteria have varied over time. In earlier years (apart from its first year, 2002), RenCon did not have a separate autonomous section—it had two sections: compulsory and open, where compulsory was limited to a fixed piano piece for all contestants and open was open to all instruments and pieces.

In these competitions, automated CSEMPs went up against human pianists and renditions which were carefully crafted by human hand. Thus, many past RenCon results are not the ideal evaluation for automated and semi-automated CSEMPs. However, they are the only published common forum available, so in the spirit of points (1) and (2) from [62], they will be referred to where possible in this chapter.

From 2008, the competition had three sections: an 'autonomous' section, a 'type-in' section and an open section. The autonomous section aims to only evaluate performances generated by automated CSEMPs. Performances are graded by the composer of the test pieces as well as by a jury. For the autonomous section, the 2008 RenCon contestants are presented with two 1-min pieces of unseen music: one in the style of Chopin and one in the style of Mozart. An award is presented for the highest-scored performance and for the performance most preferred by the composer of the two test pieces. The type-in section is for computer systems for manually generating expressive performance.

The second term of reference is polyphonic ability and refers to the ability of a CSEMP to generate expressive performances of a non-monophonic piece of music. Monophonic music has only one note playing at a time, whereas non-monophonic music can have multiple notes playing at the same time—for example, piano chords with a melody or a four-part string quartet. Many CSEMPs take monophonic inputs and generate monophonic expression, for example, SaxEx [64] focuses on a single monophonic instrument—saxophone. However, as will be seen, there are also systems like the system in Chap. 6, the ESP piano system [55] and Pop-E [28] which are designed explicitly to work with non-monophonic music. To understand why polyphonic expression is more complex than monophonic expression, consider that each voice of an ensemble may have its own melodic structure. Many monophonic methods described in our survey, if applied separately to each

part of a non-monophonic performance, could lead to parts having their own expressive timing deviations. This would cause an unsynchronised and unattractive performance [17]. Polyphonic ability is an important issue for CSEMP research because—although a significant percentage of CSEMP work has focused on monophonic expression—most music is not monophonic. Hence, Chap. 6 is dedicated to a system focusing on polyphony.

The third term of reference in this chapter, performance creativity, refers to the ability of the system to generate novel and original performances, as opposed to simulating previous human strategies. For example, the artificial neural network piano system [38] is designed to simulate human performances (an important research goal) but not to create novel performances, whereas a system like Director Musices in Chaps. 2 and 7 [17], although also designed to capture human performance strategies, has a parameterisation ability which can be creatively manipulated to generate entirely novel performances. The evolutionary computing system discussed in Chap. 4 is partially inspired by the desire for performance creativity. There is an important proviso here—a system which is totally manual would seem at first glance to have a high creativity potential, since the user could entirely shape every element of the performance. However, this potential may never be realised due to the manual effort required to implement the performance. Not all systems are able to act in a novel and practically controllable way. Many of the systems generate a model of performance which is basically a vector or matrix of coefficients. Changing this matrix by hand ('hacking it') would allow the technically knowledgeable to creatively generate novel performances. However, the changes could require too much effort or the results of such changes could be too unpredictable (thus requiring too many iterations or 'try-outs'). So performance creativity includes the ability of a system to produce novel performances with a reasonable amount of effort. Having said that, simple controllability is not the whole of performance creativity; for example, there could be a CSEMP which has only three basic performance rules which can be switched on and off with a mouse click and the new performance played immediately. However, the results of switching off and on the rules would in all likelihood generate a very uninteresting performance.

So for performance creativity, a balance needs to exist between automation and creative flexibility, since in this survey we are only concerned with automated and semi-automated CSEMPs. An example of such a balance would be an almost totally automated CSEMP but with a manageable number of parameters that can be user-adjusted before activating the CSEMP for performance. After activating the CSEMP, a performance is autonomously generated but is only partially constrained by attempting to match past human performances. Such creative and novel performance is often applauded in human performers. For example, Glenn Gould has created highly novel expressive performances of pieces of music and has been described as having a vivid musical imagination [65]. Expressive computer performance provides possibilities for even more imaginative experimentation with performance strategies.

We will now add a fourth and final dimension to the primary terms of reference which—like the other three—also has parallels in algorithmic composition. Different algorithmic composition systems generate music with different levels of structural

sophistication—for example, some may just work on the note-to-note level, like [66], whereas some may be able to plan at the higher structure level, generating forms like ABCBA, for example [67]. There is an equivalent function in computer systems for expressive performance: expressive representation. Expressive representation is the level of sophistication in the CSEMP's representation of the score as it pertains to expressive performance actions. We have already mentioned the importance of the music's structure to a human expressive performance. A piece of music can be analysed with greater and greater levels of complexity. At its simplest, it can be viewed a few notes at a time or from the point of view of melody only. At its most complex, the harmonic and hierarchical structure of the score can be analysed—as is done in Widmer's DISTALL system [49, 68]. The greater the expressive representation of a CSEMP, the more of the music features it can potentially express.

So to summarise, our four primary terms of reference will be:

- Testing status
- Expressive representation
- Polyphonic ability
- Performance creativity

At some point in the description of each system, these points will be implicitly or explicitly addressed and are summarised at the end of the chapter (see Table 1.5). It is worth noting that these are not an attempt to measure how successful the system is overall, but an attempt to highlight some key issues. What now follow are the actual descriptions of the CSEMPs, divided into a number of groups.

1.3   A Survey of Computer Systems for Expressive Music Performance

The survey presented here is meant to be representative rather than exhaustive but will cover a significant selection of published automated and semi-automated CSEMP systems to date. Each CSEMP is grouped according to how their performance knowledge is built—i.e. by learning method. This provides a manageable division of the field, showing which learning methods are most popular. The grouping will be:

1. Non-learning (10 systems)
2. Linear regression (2 systems)
3. Artificial neural networks (2 systems)
4. Rule/case-based learning (6 systems)
5. Statistical graphical models (2 systems)
6. Other regression methods (2 systems)
7. Evolutionary computation (6 systems)

The ordering of this grouping is by average year of CSEMP references within the grouping so as to help highlight trends in approaches to generating performance knowledge. For example, most early CSEMPs were non-learning, and most evolutionary computation CSEMPs have only been developed in the last few years. The fourth grouping—rule/case-based learning—is in the middle because it has been used throughout the history of CSEMP research.

1.3.1 Non-learning Systems

1.3.1.1 Director Musices

Director Musices (DM) has been an ongoing project since 1982 and is also discussed in Chaps. 2 and 7 [17, 69]. Researchers developed and tested performance rules using an analysis-by-synthesis method (later using analysis-by-measurement and studying actual performances). Currently, there are around 30 rules which are written as relatively simple equations that take as input music features such as height of the current note pitch, the pitch of the current note relative to the key of the piece or whether the current note is the first or last note of the phrase. The output of the equations defines the performance actions, for example, phrase arch which defines a 'rainbow' shape of tempo and dynamics over a phrase. The performance speeds up and gets louder towards the centre of a phrase and then tails off again in tempo and dynamics towards the end of the phrase. Some manual score analysis is required—for example, harmonic analysis and marking up of phrase start and end.

Each equation has a numeric 'k value'—the higher the k value, the more effect the rule will have and a k value of 0 switches the rule off. The results of the equations are added together linearly to get the final performance. Thanks to the adjustable k-value system, DM has much potential for performance creativity. Little work has been reported on an active search for novel performances, though it is reported that negative k values reverse rule effects and cause unusual performances. DM's ability as a semi-automated system comes from the fact that it has a 'default' set of k values, allowing the same rule settings to be applied automatically to different pieces of music (though not necessarily with the same success). Rules are also included for dealing with non-monophonic music [17].

DM is also able to deal with some performance contexts, specifically emotional expression [70], drawing on the work by Gabrielsson and Juslin [12]. Listening experiments were used to define the k-value settings on the DM rules for expressing emotions: fear, anger, happiness, sadness, solemnity, tenderness or no expression. As a result, parameters were found for each of the six rules which mould the emotional expression of a piece. A more recent development in Director Musices has been the real-time generation of performances using a version of the system called pDM [71]. Unlike pDM, many CSEMPs in this survey receive the inputs and parameters and the whole piece of music and process the data, and when this processing is complete, a generated performance is available to the user. They are not designed for real-time usage.

Director Musices has a good test status, having been evaluated in a number of experiments. In the first RenCon in 2002, the second prize went to a DM-generated performance; however, the first-placed system (a manually generated performance) was voted for by 80% of the jury. In RenCon 2005, a Director Musices default-setting (i.e. automated) performance of Mozart's Minuette KV 1(1e) came a very close 2nd in the competition, behind Pop-E [28]. However, three of the other four systems competing were versions of the DM system. The DM model has been influential, and as will be seen in the later systems, DM-type rules appear repeatedly.
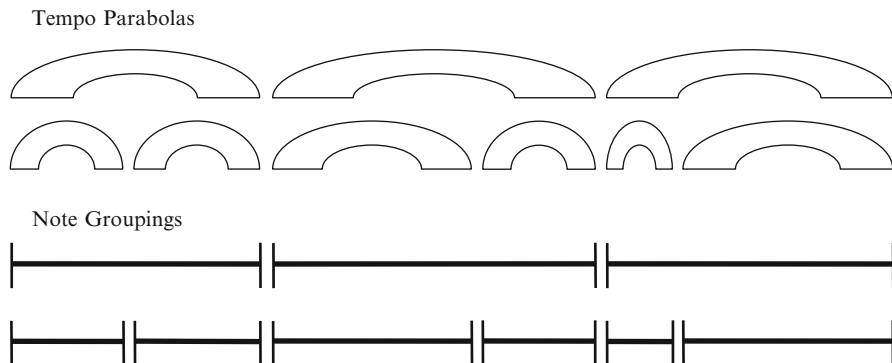
Tempo Parabolas

Note Groupings

Fig. 1.2 Todd's parabola model

1.3.1.2 Hierarchical Parabola Model

One of the first CSEMPs with a hierarchical expressive representation was Todd's hierarchical parabola model [15, 18–20]. Todd argues it was consistent with a kinematic model of expressive performance, where tempo changes are viewed as being due to accelerations and decelerations in some internal process in the human mind/body, for example, the auditory system. For tempo, the hierarchical parabola model uses a rainbow shape like DM's phrase arch, which is consistent with Newtonian kinematics. For loudness, the model uses a 'the faster the louder' rule, creating a dynamics rainbow as well.

The key difference between DM and this hierarchical model is that implicit in the hierarchical model is a greater expressive representation and wider performance action. Multiple levels of the hierarchy are analysed using Lerdahl and Jackendoff's generative theory of tonal music (GTTM). GTTM time span reduction (TSR) examines each note's musicological place in all hierarchical levels. The rainbows/parabolas are generated at each level, from the note-group level upwards (Fig. 1.2), and added to get the performance. This generation is done by a parameterised parabolic equation which takes as input the result of the GTTM TSR analysis.

The performance was shown to correlate well by eye with a short human performance, but no correlation figures were reported. Clarke and Windsor (2000) tested the first four bars of Mozart's K.331, comparing two human performers with two performances by the hierarchical parabola model. Human listeners found the parabola version unsatisfactory compared to the human ones. In the same experiment however, the parabola model was found to work well on another short melody. The testing also showed that the idea of 'the louder the faster' did not always hold. Desain and Honing [72] claim, as a result of informal listening tests, that in general the performances do not sound convincing.

The constraint of utilising the hierarchy and the GTTM TSR approach limits the performance creativity. Note groupings will be limited to those generated by a GTTM TSR analysis, and the parabolas generated will be constrained by the model's equation. Any adjustments to a performance will be constrained to working within this framework.

Table 1.4 Level 2 composers' pulses

Level 2 composers' pulses—four pulses					
Beethoven	Duration	106	89	96	111
	Amplitude	1.00	0.39	0.83	0.81
Mozart	Duration	105	95	105	95
	Amplitude	1.00	0.21	0.53	0.23
Schubert	Duration	97	114	98	90
	Amplitude	1.00	0.65	0.40	0.75
Haydn	Duration	108	94	97	102
	Amplitude	1.00	0.42	0.68	1.02
Schumann	Duration	96	116	86	102
	Amplitude	0.60	0.95	0.50	1.17
Mendelssohn	Duration	118	81	95	104
	Amplitude	1.00	0.63	0.79	1.12
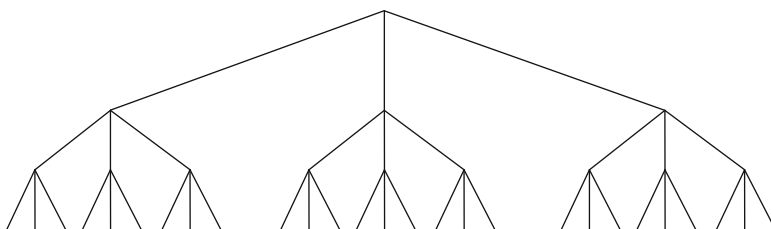
Fig. 1.3 Structure of a pulse set in three-time

1.3.1.3 Composer Pulse and Predictive Amplitude Shaping

Manfred Clynes' composer pulse [21] also acts on multiple levels of the hierarchy. Clynes hypothesises that each composer has a unique pattern of amplitude and tempo variations running through performances—a pulse. This is captured as a set of numbers multiplying tempo and dynamics values in the score. It is hierarchical with separate values for within the beat, the phrase and at multiple bar level. Table 1.4 shows the values of pulses for phrase level for some composers. The pulses were measured using a sentograph to generate pressure curves from musicians tapping their finger whilst thinking of or listening to a specific composer. Figure 1.3 shows the structure of a pulse set in three-time (each composer has a three-time and a four-time pulse set defined). This pulse set is repeatedly applied to a score end on end. So if the pulse is 12 beats long and the score is 528 beats, the pulse will repeat 528/12 = 44 times end on end.

Another key element of Clynes' approach is predictive amplitude shaping. This adjusts a note's dynamics based on the next note, simulating 'a musician's unconscious ability to sculpt notes in this way' that 'makes his performance flow beautifully through time, and gives it meaningful coherence even as the shape and duration of each individual note is unique'. A fixed envelope shape model is used (some constants are manually defined by the user), the main inputs being distant to

the next note and duration of the current note. So the pulse/amplitude system has only note-level expressive representation.

Clynes' test of his own model [22] showed that a number of expert and non-expert listeners preferred music with a composer's pulse than with a different pulse. However, not all tests on Clynes' approach have supported a universal pulse for each composer [73, 74], suggesting instead that the pulse may be effective for a subset of a composer's work. Clynes' pulses and amplitude shaping have been combined with other performance tools (e.g. vibrato generation) as part of his commercial software SuperConductor. Two SuperConductor-generated performances were submitted to RenCon 2006 open section: Beethoven's Eroica Symphony, Op.55, Mvt.4 and Brahms' Violin Concerto, Op.77, Mvt.1. The Beethoven piece scored low, but the Brahms piece came first in the open section (beating two pieces submitted by Pop-E [28]). The generation of this piece could have involved significant amounts of manual work. Also because it was the open section, the pieces submitted by Pop-E were not the same as submitted by SuperConductor—hence, like was not compared to like. SuperConductor also won the open section in RenCon 2004 with J. S. Bach, Brandenburg Concerto No. 5, D Major, 3rd Movement. The only competitor included from this survey was Rubato [26] performing a Bach piece. It should be re-emphasised that these results were for SuperConductor and not solely for the pulse and amplitude tools.

In the context of SuperConductor, Clynes' approach allows for significant performance creativity. The software is designed to allow a user to control the expressive shaping of a MIDI performance, giving significant amounts of control. However, outside of the context of SuperConductor, the pulse has little scope for performance creativity—though the amplitude shaping does. The pulse and amplitude shaping do not explicitly address non-monophonic music, though SuperConductor can be used to generate polyphonic performances.

1.3.1.4 Bach Fugue System

In the Bach fugue system [23], expert system methods are used to generate performance actions. Johnson generated the knowledge base through interviews with two musical expert performers and through a performance practice manual and an annotated edition of the Well-Tempered Clavier; so this system is not designed for performance creativity. Twenty-eight conditions for tempo and articulation are so generated for the knowledge base. For example, 'If there is any group of 16th notes following a tied note, then slur the group of 16th notes following the long note'. Expressive representation is focused on the note to phrase level. The CSEMP does not perform itself but generates instructions for 4/4 fugues. So testing was limited to examining the instructions. It gave the same instructions as human experts 85–90% of the time, though it is not said how many tests were run. The system is working in the context of polyphony.

1.3.1.5 Trumpet Synthesis

The testing of many of the CSEMPs surveyed has focused on keyboard. This pattern will continue through the chapter—most historical CSEMPs focused on the piano because it was easier to collect and analyse data for the piano than

for other instruments. However, this book shows that this trend is starting to change (e.g. see Chaps. 5 and 9). One of the first non-piano systems was Dannenberg and Derenyi's trumpet synthesis [24, 25]. The authors' primary interest here was to generate realistic trumpet synthesis, and adding performance factors improves this synthesis. It is not designed for performance creativity but for simulation. This trumpet system synthesises the whole trumpet performance, without needing any MIDI or audio building blocks as the basis of its audio output. The performance actions are amplitude and frequency, and these are controlled by envelope models which were developed using a semi-manual statistical analysis-by-synthesis method. A 10-parameter model was built for amplitude, based on elements such as articulation, direction and magnitude of pitch intervals and duration of notes. This system works by expressively transforming one note at a time, based on the pattern of the surrounding two notes. In terms of expressive representation, the system works on a three-note width. The pitch expression is based on envelopes which were derived and stored during the analysis-by-synthesis.

No test results are reported. Dannenberg and Derenyi placed two accompanied examples online: parts of a Haydn Trumpet Concerto and of a Handel Minuet. The start of the trumpet on the concerto without accompaniment is also online, together with a human playing the same phrase. The non-accompanied synthesis sounds quite impressive, only being let down by a synthetic feel towards the end of the phrase—though the note-to-note expression (as opposed to the synthesis) consistently avoids sounding machine-like. In both accompanied examples, it became clear as the performances went on that a machine was playing, particularly in faster passages. But once again, note-to-note expression did not sound too mechanical. Despite the reasonably positive nature of these examples, there is no attempt to objectively qualify how good the trumpet synthesis system is.

1.3.1.6 Rubato

Mazzola, a mathematician and recognised jazz pianist, developed a mathematical theory of music [26, 27]. Music is represented in an abstract geometrical space whose coordinates include onset time, pitch and duration. A score will exist in this space, and expressive performances are generated by performing transformations on the space. The basis of these transformations is a series of 'operators' which can be viewed as a very generalised version of the rule-based approach taken in Director Musices. For example, the tempo operator and the split operator allow the generation of tempo hierarchies. These give rubato a good expressive representation. However, the definition of the hierarchy here differs somewhat from that found in the hierarchical parabola model [18, 19] or DISTALL [50, 51]. A tempo hierarchy, for a piano performance, may mean that the tempo of the left hand is the dominant tempo, at the top of a hierarchy, and the right-hand tempo is always relative to the left-hand tempo—and so is viewed as being lower in the hierarchy. Mazzola also discusses the use of tempo hierarchies to generate tempo for grace notes and arpeggios—the tempo of these is relative to some global tempo higher in the hierarchy. Ideas from this theory have been implemented in a piece of software called Rubato, which is available online. The expressive performance module in

Rubato is the 'Performance Rubette'. A MIDI file can be loaded in Rubato and predefined operators used to generate expressive performances. The user can also manually manipulate tempo curves using a mouse and GUI, giving Rubato good scope for performance creativity.

Test reports are limited. In RenCon 2004, a performance of Bach's Contrapunctus III modelled using Rubato was submitted and came fourth in the open section (SuperConductor came first in the section with a different piece). It is not clear how automated the generation of the performance was. Listening to the submission, it can be heard that although the individual voices are quite expressive and pleasant (except for the fastest parts), the combination sounds relatively unrealistic. An online MIDI example is available of Schumann's Kindersezenen Op. 15 No. 2, 'Kuriose Geschichte', which evidences both tempo and dynamics expression and is quite impressive, though once again it is not clear how automated the production of the music was.

1.3.1.7 Pop-E

Pop-E [28], a polyphrase ensemble system, was developed by some of the team involved in MIS [32, 33]. It applies expression features separately to each voice in a MIDI file, through a synchronisation algorithm. The music analysis uses GTTM local level rules and utilises beams and slurs in the score to generate note groupings. So the expressive representation is up to phrase level. Expressive actions are applied to these groupings through rules reminiscent of Director Musices. The five performance rules have a total of nine manual parameters between them. These parameters can be adjusted, providing scope for performance creativity. In particular, jPop-E [75], a java implementation of the system, provides such tools for shaping new performances.

To deal with polyphony, synchronisation points are defined at the note-grouping start and end points in the attentive part. The attentive part is the voice which is most perceptually prominent to a listener. The positions of notes in all other non-attentive parts are linearly interpolated relative to the synchronisation points (defined manually). This means that all parts will start and end at the same time at the start and end of groupings of the main attentive part.

Pop-E was evaluated in the laboratory to see how well it could reconstruct specific human performances. After setting parameters manually, performances by three pianists were reconstructed. The average correlation values between Pop-E and a performer were 0.59 for tempo and 0.76 for dynamics. This has to be viewed in the context that the average correlations between the human performers were 0.4 and 0.55, respectively. Also, the upper piano part was more accurate on average. (It is interesting to note that for piano pieces whose attentive part is the right hand, the Pop-E synchronisation system is similar to the methods in the DISTALL [50] system for dealing with polyphony.) Pop-E won the RenCon 2005 compulsory section, beating Director Musices [17] (Chaps. 2 and 7). In RenCon 2006, Pop-E won the compulsory section, beating Kagurame [43] and Ha-Hi-Hun [45]. In the open section in 2006, SuperConductor [21, 22] beat Pop-E with one performance and lost to Pop-E with another.

1.3.1.8 Hermode Tuning

The next two subsections describe successful commercial CSEMPs. Despite the lack of details available on these proprietary systems, they should be included here, since they are practical CSEMPs that people are paying money for and illustrate some of the commercial potential of CSEMPs for the music business. However, because of the lack of some details, the four review terms of reference will not be applied. The first system is Hermode tuning [29]. Most systems in this chapter focus on dynamics and timing. However, intonation is another significant area of expression for many instruments—for example, many string instruments. (In fact, three intonation rules were added to Director Musices in its later incarnations, e.g. the higher the pitch, the sharper the note.) Hermode tuning is a dedicated expressive intonation system which can work in real time, its purpose being to 'imitate the living intonation of well-educated instrumentalists in orchestras and chamber music ensembles'. Instrumentalists do not perform in perfect intonation—in fact, if an orchestra performed music in perfect tuning all the time, the sound would be less pleasant than one that optimised its tuning through performance experience. A series of algorithms are used in Hermode tuning, not just to avoid perfect intonation but to attempt to achieve optimal intonation. The algorithms have settings for different types of music, for example, baroque and jazz/pop. Examples are available on the website, and the system has been successful enough to be embedded in a number of commercial products—for example, Apple Logic Pro 7.

1.3.1.9 Sibelius

As mentioned in the introduction of this chapter, the music typesetting software package Sibelius has built-in algorithms for expressive performance. These use a rule-based approach. Precise details are not available for these commercial algorithms, but some information is available [30]. For dynamics, beat groups such as bar lines, sub-bar groups and beams are used to add varying degrees of stress. Also, the higher the note is, the louder it is played, though volume resets at rests and dynamics expression is constrained to not be excessive. Some random fluctuation is added to dynamics to make it more human sounding as well. Tempo expression is achieved using a simple phrase-based system; however, this does not include reliable phrase analysis. The manufacturer reports that 'phrasing need only be appropriate perhaps 70% of the time—the ear overlooks the rest' and that 'the ear is largely fooled into thinking it is a human performance'. Notion and Finale also have expressive performance systems built into them, which are reportedly more advanced than Sibelius', but even fewer details are available for the proprietary methodologies in these systems.

1.3.1.10 Computational Music Emotion Rule System

In relation to the philosophy behind the computational music emotion rule system (CMERS) [31], Livingstone 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 computational music emotion rule system has a rule set of 19 rules developed through analysis-by-synthesis. The rules

have an expressive representation up to the phrase level, some requiring manual markup of the score. 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 with four quadrants going from very active and negative to very active and positive, to very passive and positive through very passive and negative. These four elements combine to give such emotions as angry, bright, contented and despairing. The quadrants were constructed from a review of 20 studies of music and emotion. The rules for expressing emotions include moving between major and minor modes, changing note pitch classes and DM-type rules for small changes in dynamics and tempo. It was found that the addition of the microfeature humanisation rules improved the accuracy of the emotional expression (as opposed to solely using macrofeature 'recomposition' rules). The rules for humanising the performance include some rules which are similar to Director Musices, such as phrase arch and emphasising metrically important beats. Creative performance is possible in CMERS by adjusting the parameters of the rule set, and the emotional specification would allow a user to specify different emotions for different parts of a performance.

A significant number of formal listening tests have been done by Livingstone, and they support the hypothesis that CMERS is more successful than DM at expressing emotions. CMERS is one of the better tested systems in this chapter—one reason being that its aim is more measurable than a purely aesthetic goal. Examples of CMERS are available on the author's webpage.

1.3.2 Linear Regression

Learning CSEMPs can incorporate more knowledge more quickly than non-learning systems. However, such methods do not always provide tools for creative performance because they are strongly rooted in past performances. Before continuing, it should be explained that any CSEMP that learns expressive deviations needs to have a non-expressive reference point, some sort of representation of the music played robotically/neutrally. The CSEMP can then compare this to the score played expressively by a human and learn the deviations. Linear regression is the first learning method which will be addressed. Linear regression models assume a basically linear relationship between the music features and the expressive actions. The advantage of such models is their simplicity. The disadvantage is that assuming music expressive performance as a linear process is almost certainly an oversimplification.

1.3.2.1 Music Interpretation System

The music interpretation systems (MIS) [32–34] generate expressive performances in MIDI format but learn expressive rules from audio recordings. This is done using a spectral analysis system with dynamic programming for note detection.

The system is a simulatory CSEMP and uses a set of linear equations which map score features onto performance deviation actions. Its expressive representation is on the note and phrase level. MIS has methods to include some non-linearities using logical ANDs between music features in the score and a way of reducing redundant music features from its equations. This redundancy reduction improves 'generalisation' ability (the ability for the system to perform music or composers that were not explicitly included in its learning). MIS learns links between music features and performance actions of tempo, dynamics and articulation. The music features used include score expression marks and aspects of GTTM and two other forms of musicological analysis: Leonard Meyer's theory of musical meaning [76] and Narmour's IR theory. IR considers features of the previous two notes in the melody and postulates that a human will expect the melody to move in a certain direction and distance; thus, it can classify each note as being part of a certain expectation structure. Meyer's theory is also an expectation-based approach, but coming from the perspective of game theory.

For testing, MIS was trained on the first half of a Chopin waltz and then used to synthesise the second half. Correlations (accuracies when compared to a human performance of the second half) were: for velocity 0.87, for tempo 0.75 and for duration 0.92. A polyphonic MIS interpretation of Chopin Op. 64 No. 2 was submitted to RenCon 2002. It came third behind DISTALL [50], beating three of the other four automated systems—DM, Kagurame [43] and Ha-Hi-Hun [45].

1.3.2.2 CaRo

CaRo [35–37, 77, 78] is a monophonic CSEMP designed to generate audio files which—like CMERS [31]—express certain moods/emotions. It does not require a score to work from but works on audio files which are mood neutral. The files are however assumed to include the performer's expression of the music's hierarchical structure. Its expressive representation is at the local note level. CaRo's performance actions at the note and intranote level include changes to inter-onset interval, brightness and loudness-envelope centroid. A linear model is used to learn actions—every action has an equation characterised by parameters called shift and range expansion. Each piece of music in a particular mood has its own set of shift and range expansion values. This limits the generalisation potential.

CaRo also learns 'how musical performances are organised in the listener's mind' in terms of moods: hard, heavy, dark, bright, light and soft. To do this, a set of listening experiments analysed by principal component analysis (PCA) generate a two-dimensional space that captures 75% of the variability present in the listening results; this space is used to represent listeners' experience of the moods. A further linear model is learned for each piece of music which maps the mood space onto shift and range expansion values. The user can select any point in the mood space, and CaRo generates an expressive version of the piece. A line can be drawn through mood space, and following that line, in time CaRo can generate a performance morphing through different moods. Apart from the ability to adjust shift and range expansion parameters manually, CaRo's potential for creative performance is

extended by its ability to have a line drawn through the mood space. Users can draw trajectories through this space which create entirely novel performances; and this can be done in real time.

For testing, 20-s clips, each from three piano pieces by different composers, were used. A panel of 30 listeners evaluated CaRo's ability to generate pieces with different expressive moods. Results showed that the system gave a good modelling of expressive mood performances as realised by human performers.

1.3.3 Artificial Neural Networks

1.3.3.1 Artificial Neural Network Piano System

The earliest ANN approach is the artificial neural network piano system [38]. It has two incarnations. The first did not learn from human performers: a set of seven monophonic Director Musices rules were selected, and two (loudness and timing) feedforward ANNs learned these rules through being trained on them. By learning a fixed model of Director Musices, the ANN loses the performance creativity of the k values. When monophonic listening tests were done with 20 subjects, using Mozart's Piano Sonatas K331 and K281, the Director Musices performance was rated above the non-expressive computer performance, but the neural network performance rated highest of all. One explanation for the dominance of the ANN over the original DM rules was that the ANN generalised in a more pleasant way than the rules. The other ANN system by Bresin was a simulation CSEMP which also used a separate loudness and timing feedback ANN. The ANNs were trained using actual pianist performances from MIDI, rather than on DM rules, but some of the independently learned rules turned out to be similar to some DM rules. Informal listening tests judged the ANNs as musically acceptable. The network looked at a context of four notes (loudness) and five notes (timing) and so had note- to phrase-level expressive representation, though it required the notes to be manually grouped into phrases before being input.

1.3.3.2 Emotional Flute

The emotional flute system [39] uses explicit music features and artificial neural networks, thus allowing greater generalisation than the related CaRo system [35, 36]. The music features are similar to those used in Director Musices. This CSEMP is strongly related to Bresin's second ANN, extending it into the non-piano realm and adding mood space modelling. Expressive actions include inter-onset interval, loudness and vibrato. Pieces need to be segmented into phrases before being input—this segmentation is performed automatically by another ANN. There are separate nets for timing and for loudness—net designs are similar to Bresin's and have similar levels of expressive representation. There is also a third net for the duration of crescendo and decrescendo at the single note level. However, the nets could not be successfully trained on vibrato, so a pair of rules were generated to handle it. A flautist performed the first part of Telemann's Fantasia No. 2 in nine different moods: cold, natural, gentle, bright, witty, serious, restless, passionate and dark. Like

CaRo, a 2-D mood space was generated and mapped onto the performances by the ANNs, and this mood space can be utilised to give greater performance creativity.

To generate new performances, the network drives a physical model of a flute. Listening tests gave an accuracy of approximately 77% when subjects attempted to assign emotions to synthetic performances. To put this in perspective, even when listening to the original human performances, human recognition levels were not always higher than 77%; the description of emotional moods in music is a fairly subjective process.

1.3.4 Case and Instance-Based Systems

1.3.4.1 SaxEx

Arcos and Lopez de Mantaras' SaxEx [40–42, 64, 79] was one of the first systems to learn performances based on the performance context of mood. Like the trumpet system described earlier [24, 25], SaxEx includes algorithms for extracting notes from audio files and generating expressive audio files from note data. SaxEx also looks at intranote features like vibrato and attack. Unlike the trumpet system, SaxEx needs a non-expressive audio file to perform transformations upon. Narmour's IR theory is used to analyse the music. Other elements used to analyse the music are ideas from jazz theory as well as GTTM TSR. This system's expressive representation is up to phrase level and is automated.

SaxEx was trained on cases from monophonic recordings of a tenor sax playing four jazz standards with different moods (as well as a non-expressive performance). The moods were designed around three dimensions: tender-aggressive, sad-joyful and calm-restless. The mood and local IR, GTTM and jazz structures around a note are linked to the expressive deviations in the performance of that note. These links are stored as performance cases. SaxEx can then be given a non-expressive audio file and told to play it with a certain mood. A further AI method is used then to combine cases: fuzzy logic. For example, if two cases are returned for a particular note in the score and one says play with low vibrato and the other says play with medium vibrato, then fuzzy logic combines them into a low-medium vibrato. The learning of new CBR solutions can be done automatically or manually through a GUI, which affords some performance creativity giving the user a stronger input to the generation of performances. However, this is limited by SaxEx's focus on being a simulation system. There is—like the computational music emotion rule system [31]—the potential for the user to generate a performance with certain moods at different points in the music.

There is no formal testing reported, but SaxEx examples are available online. The authors report 'dozens' of positive comments about the realism of the music from informal listening tests, but no formal testing is reported or details given. The two short examples online (sad and joyful) sound realistic to us, more so than, for example, the trumpet system examples. But the accuracy of the emotional expression was difficult for us to gauge.

1.3.4.2 Kagurame

Kagurame [43, 44] is another case-based reasoning system which—in theory—also allows expressiveness to be generated from moods, this time for piano. However, it is designed to incorporate a wider degree of performance conditions than solely mood, for example, playing in a baroque or romantic style. Rather than GTTM and IR, Kagurame uses its own custom hierarchical note structures to develop and retrieve cases for expressive performance. This hierarchical approach gives good expressive representation. Score analysis automatically divides the score into segments recursively with the restriction that the divided segment must be shorter than one measure. Hence, manual input is required for boundary information for segments longer than one measure. The score patterns are derived automatically after this, as is the learning of expressive actions associated with each pattern. Kagurame acts on timing, articulation and dynamics. There is also a polyphony action called chord time lag—notes in the same chord can be played at slightly different times. It is very much a simulation system with little scope for creative performance.

Results are reported for monophonic classical and romantic styles. Tests were based on learning 20 short Czerny etudes played in each style. Then a 21st piece was performed by Kagurame. Listeners said it 'sounded almost human-like, and expression was acceptable' and that the 'generated performance tended to be similar to human, particularly at characteristic points'. A high percentage of listeners guessed correctly whether the computer piece was romantic or classical style. In RenCon 2004, Kagurame came fourth in one half of the compulsory section, one ahead of Director Musices, but was beaten by DM in the second half, coming fifth. At RenCon 2006, a polyphonic performance of Chopin's piano Etude in E major came second—with Pop-E [28] taking first place.

1.3.4.3 Ha-Hi-Hun

Ha-Hi-Hun [45] utilises data structures designed to allow natural language statements to shape performance conditions (these include data structures to deal with non-monophonic music). The paper focuses on instructions of the form 'generate performance of piece X in the style of an expressive performance of piece Y'. As a result, there are significant opportunities for performance creativity through generating a performance of a piece in the style of a very different second piece or perhaps performing the Y piece, bearing in mind that it will be used to generate creative performances of the X piece. A similar system involving some of the same researchers is discussed in Chap. 8 as part of introducing an approach for automated music structure analysis. The music analysis of Ha-Hi-Hun uses GTTM TSR to highlight the main notes that shape the melody. TSR gives Ha-Hi-Hun an expressive representation above note level. The deviations of the main notes in the piece Y relative to the score of Y are calculated and can then be applied to the main notes in the piece X to be performed by Ha-Hi-Hun. After this, the new deviations in X's main notes are propagated linearly to surrounding notes like 'expressive ripples' moving outwards. The ability of Ha-Hi-Hun to automatically generate expressive performances comes from its ability to generate a new performance X based on a previous human performance Y.

In terms of testing, performances of two pieces were generated, each in the style of performances of another piece. Formal listening results were reported as positive, but few experimental details are given. In RenCon 2002, Ha-Hi-Hun learned to play Chopin Etude Op. 10 No. 3 through learning the style of a human performance of Chopin's Nocturne Op. 32 No. 2. The performance came ninth out of ten submitted performances by other CSEMPs (many of which were manually produced). In RenCon 2004, Ha-Hi-Hun came last in the compulsory section, beaten by both Director Musices and Kagurame [43]. In RenCon 2006, a performance by Ha-Hi-Hun also came third out of six in the compulsory section, beaten by Pop-E [28] and Kagurame.

1.3.4.4 PLCG System

Gerhard Widmer has applied various versions of a rule-based learning approach, attempting to utilise a larger database of music than previous CSEMPs. Chapter 3 discusses some of this work. The PLCG system [46–48] uses data mining to find large numbers of possible performance rules and cluster each set of similar rules into an average rule. This is a system for musicology and simulation rather than one for creative performance. PLCG is Widmer's own meta-learning algorithm—the underlying algorithm being sequential covering [80]. PLCG runs a series of sequential covering algorithms in parallel on the same monophonic musical data and gathers the resulting rules into clusters, generating a single rule from each cluster. The data set was 13 Mozart piano sonatas performed by Roland Batik in MIDI form (only melodies were used—giving 41,116 notes). A note-level structure analysis learns to generate tempo, dynamics and articulation deviations based on the local context—for example, size and direction of intervals, durations of surrounding notes and scale degree. So this CSEMP has a note-level expressive representation. As a result of the PLCG algorithm, 383 performance rules were turned into just 18 rules. Interestingly, some of the generated rules had similarities to some of the Director Musices rule set.

Detailed testing has been done on the PLCG, including its generalisation ability. Widmer's systems are the only CSEMPs surveyed in this chapter that have had any significant generalisation testing. The testing methods were based on correlation approaches. Seven pieces selected from the scores used in learning were regenerated using the rule set. Their tempo/dynamics profiles compared very favourably to those of the original performances. Regenerations were compared to performances by a different human performer Philippe Entremont and showed no degradation relative to the original performer comparison. The rules were also applied to some music in a romantic style (two Chopin pieces), giving encouraging results. There are no reports of formal listening tests.

1.3.4.5 Combined Phrase-Decomposition/PLCG

The above approach was extended by Widmer and Tobudic into a monophonic system whose expressive representation extends into higher levels of the score hierarchy. This was the combined phrase-decomposition/PLCG system [49]. Once again, this is a simulation system rather than one for creative performance.

When learning, this CSEMP takes as input scores that have had their hierarchical phrase structure defined to three levels by a musicologist (who also provides some harmonic analysis), together with an expressive MIDI performance by a professional pianist. Tempo and dynamics curves are calculated from the MIDI performance, and then the system does a multilevel decomposition of these expression curves. This is done by fitting quadratic polynomials to the tempo and dynamics curves (similar to the curves found in Todd's parabola model [18, 19]).

Once the curve fitting has been done, there is still a 'residual' expression in the MIDI performance. This is hypothesised as being due to note-level expression, and the PLCG algorithm is run on the residuals to learn the note-level rules which generate this residual expression. The learning of the non-PLCG tempo and dynamics is done using a case-based learning type method—by a mapping from multiple-level features to the parabola/quadratic curves. An extensive set of music features are used, including length of the note group, melodic intervals between start and end notes, where the pitch apex of the note group is, whether the note group ends with a cadence and the progression of harmony between start, apex and end. This CSEMP has the most sophisticated expressive representation of all the systems described in this chapter.

To generate an expressive performance of a new score, the system moves through the score and in each part runs through all its stored musical features vectors learned from the training; it finds the closest one using a simple distance measure. It then applies the curve stored in this case to the current section of the score. Data for curves at different levels and results of the PLCG are added together to give the expression performance actions.

A battery of correlation tests were performed. Sixteen Mozart sonatas were used to test the system—training on 15 of them and then testing against the remaining one. This process was repeated independently, selecting a new 1 of the 16 and then retraining on the other 15. This gave a set of 16 results which the authors described as 'mixed'. Dynamics generated by the system correlated better with the human performance than a non-expressive performance curve (i.e. straight line) did, in 11 out of 16 cases. For the timing curves, this was true for only 6 out of 16 cases. There are no reports of formal listening tests.

1.3.4.6 DISTALL System

Widmer and Tobudic did further work to improve the results of the combined phrase-decomposition/PLCG, developing the DISTALL system [50, 51] for simulation. The learned performance cases in the DISTALL system are hierarchically linked, in the same way as the note groupings they represent. So when the system is learning sets of expressive cases, it links together the feature sets for a level 3 grouping with all the level 2 and level 1 note groupings it contains. When a new piece is presented for performance and the system is looking at a particular level 3 grouping of the new piece, say X—and X contains a number of level 2 and level 1 subgroupings—then not only are the score features of X compared to all level 3 cases in the memory but the subgroupings of X are compared to the subgroupings of the compared level 3 cases as well. There have been measures

available which can do such a comparison in case-based learning before DISTALL (e.g. RIBL [81]). However, DISTALL does it in a way more appropriate to expressive performance—giving a more equal weighting to subgroupings within a grouping and giving this system a high expressive representation.

Once again, correlation testing was done with a similar set of experiments to the section above. All 16 generated performances had smaller dynamics errors relative to the originals than a robotic/neutral performance had. For tempo, 11 of the 16 generated performances were better than a robotic/neutral performance. Correlations varied from 0.89 for dynamics in Mozart K283 to 0.23 for tempo in Mozart K332. The mean correlation for dynamics was 0.7 and for tempo was 0.52. A performance generated by this DISTALL system was entered into RenCon 2002. The competition CSEMP included a simple accompaniment system where dynamics and timing changes calculated for the melody notes were interpolated to allow their application to the accompaniment notes as well. Another addition was a simple heuristic for performing grace notes: the sum of durations of all grace notes for a main note is set equal to 5% of the main note's duration, and the 5% of duration is divided equally amongst the grace notes. The performance was the top-scored automated performance at RenCon 2002—ahead of Kagurame [43], MIS [32] and Ha-Hi-Hun [45]—and it beat one non-automated system.

1.3.5 Statistical Graphical Models

1.3.5.1 Music Plus One

The Music Plus One system [52–54] is able to deal with multiple-instrument polyphonic performances. It has the ability to adjust performances of polyphonic sound files (e.g. orchestral works) to fit as accompaniment for solo performers. This CSEMP contains two modules: the listen and play modules. Listen uses a hidden Markov model (HMM) to track live audio and find the soloist's place in the score in real time. Play uses a Bayesian belief network (BBN) which, at any point in a soloist performance and based on the performance so far, tries to predict the timing of the next note the soloist will play. Music Plus One's BBN is trained by listening to the soloist. As well as timing, the system learns the loudness for each phrase of notes. However, loudness learning is deterministic—it performs the same for each accompaniment of the piece once trained, not changing based on the soloist changing his or her own loudness. Expressive representation is at the note level for timing and phrase level for loudness.

The BBN assumes a smooth changing in tempo, so any large changes in tempo (e.g. a new section of a piece) need to be manually marked up. For playing MIDI files for accompaniment, the score needs to be divided up manually into phrases for dynamics; for using audio files for accompaniment, such a division is not needed. When the system plays back the accompaniment, it can play it back in multiple expressive interpretations depending on how the soloist plays. So it has learned a flexible (almost tempo independent) concept of the soloist's expressive intentions for the piece.

There is no test reported for this system—the author stated the impression that the level of musicality obtained by the system is surprisingly good and asked readers to evaluate the performance themselves by going to the website and listening. Music Plus One is actually being used by composers and for teaching music students. It came in first at RenCon 2003 in the compulsory section with a performance of Chopin's Prelude Number 15, Raindrop, beating Ha-Hi-Hun [45], Kagurame [43] and Widmer's system. To train Music Plus One for this, several performances were recorded played by a human, using a MIDI keyboard. These were used to train the BBN. The model was extended to include velocities for each note, as well as times, with the assumption that the velocity varies smoothly (like a random walk) except at hand-identified phrase boundaries. Then a mean performance was generated from the trained model.

As far as performance creativity goes, the focus on this system is not so much to generate expressive performances as to learn the soloist's expressive behaviour and react accordingly in real time. However, the system has an 'implicit' method of creating new performances of the accompaniment—the soloist can change his or her performance during playback. There is another creative application of this system: multiple pieces have been composed for use specifically with the Music Plus One system—pieces which could not be properly performed without the system. One example contains multiple sections where one musician plays 7 notes whilst the other plays 11. Humans would find it difficult to do this accurately, whereas a soloist and the system can work together properly on this complicated set of polyrhythms.

1.3.5.2 ESP Piano System

Grindlay's ESP piano system [55] is a polyphonic CSEMP designed to simulate expressive playing of pieces of piano music which consist of a largely monophonic melody, with a set of accompanying chords (known as homophony). A hidden Markov model learns expressive performance using music features such as whether the note is the first or last of the piece, the position of the note in its phrase and the note duration relative to its start and the next note's start (called its 'articulation' here). The expressive representation is up to the phrase level. Phrase division is done manually, though automated methods are discussed. The accompaniment is analysed for a separate set of music features, some of which are like the melody music features. Some are unique to chords—for example, the level of consonance/dissonance of the code (based on a method called Euler's solence). Music features are then mapped onto a number of expressive actions such as (for melody) the duration deviation and the velocity of the note compared to the average velocity. For the accompaniment, similar actions are used as well as some chord-only actions, like the relative onset of chord notes (similar to the Kagurame chord time lag). These chordal values are based on the average of the values for the individual notes in the chord.

Despite the focus of this system on homophony, tests were only reported for monophonic melodies, training the HMM on 10 graduate performances of Schumann's Träumerei. Ten out of 14 listeners ranked the expressive ESP output over the inexpressive version. Ten out of 14 ranked the ESP output above that of

an undergraduate performance. Four out of seven preferred the ESP output to a graduate student performance.

1.3.6   Other Regression Methods

1.3.6.1 Drumming System

Thus far in this chapter, only pitched instrument systems have been surveyed—mainly piano, saxophone and trumpet. A system for non-pitched (drumming) expression will now be examined. In the introduction, it was mentioned that pop music enthusiastically utilised the 'robotic' aspects of MIDI sequencers. However, eventually pop musicians wanted a more realistic sound to their electronic music, and humanisation systems were developed for drum machines that added random tempo deviations to beats. Later systems also incorporated what are known as grooves—a fixed pattern of tempo deviations which are applied to a drum beat or any part of a MIDI sequence (comparable to a one-level Clynes pulse set discussed earlier). Such groove systems have been applied commercially in mass-market systems like Propellerhead Reason, where it is possible to generate groove templates from a drum track and apply it to any other MIDI track [56]. However, just as some research has suggested limitations in the application of Clynes' composer pulses, so Wright and Berdahl's [82] research shows the limits of groove templates. Their analysis of multi-voiced Brazillian drumming recordings found that groove templates could only account for 30% of expressive timing.

Wright and Berdahl investigated other methods to capture the expressive timing using a system that learns from audio files. The mapping model is based on machine learning regression between audio features and timing deviation (versus a quantized version of the beat). Three different methods of learning the mapping model were tried. This learning approach was found to track the expressive timing of the drums much better than the groove templates. Examples are provided online. Note that the system is not only limited to Brazilian drumming; Wright and Berdahl also tested it on reggae rhythms with similar success.

1.3.6.2 KCCA Piano System

The most recent application of kernel regression methods to expressive performance is the system by [57]. Their main aim is simulatory, to imitate the style of a particular performer and allow new pieces to be automatically performed using the learned characteristics of the performer. A performer is defined based on the 'worm' representation of expressive performance [83]. The worm is a visualisation tool for the dynamics and tempo aspects of expressive performance. It uses a 2-D representation with tempo on the x-axis and loudness on the y-axis. Then, as the piece plays, at fixed periods in the score (e.g. once per bar), an average is calculated for each period and a filled circle plotted on the graph at the average. Past circles remain on the graph, but their colour fades and size decreases as time passes—thus creating the illusion of a wriggling worm whose tail fades off into the distance in time. If the computer played an expressionless MIDI file, then its worm would stand still, not wriggling at all.

The basis of Dorard's approach is to assume that the score and the human performances of the score are two views of the musical semantic content, thus enabling a correlation to be drawn between the worm and the score. The system focuses on homophonic piano music—a continuous upper melody part and an accompaniment—and divides the score into a series of chord and melody pairs. Kernel canonical correlation analysis (KCCA) [84] is then used, a method which looks for a common semantic representation between two views. Its expressive representation is based on the note-group level, since KCCA is looking to find correlations between short groups of notes and the performance worm position. An addition needed to be made to the learning algorithm to prevent extreme expressive changes in tempo and dynamics. This issue is a recurring problem in a number of CSEMPs (e.g. see the systems discussed in this chapter—artificial neural network models and Sibelius).

Testing was performed on Chopin's Etude Op. 10 No. 3—the system was trained on the worm of the first 8 bars and then tried to complete the worm for bars 9–12. The correlation between the original human performance worm for 9–12 and the reconstructed worm was measured to be 0.95, whereas the correlation with a random worm was 0.51. However, the resulting performances were reported (presumably through informal listening tests) to be not very realistic.

1.3.7 Evolutionary Computation

A number of more recent CSEMPs have used evolutionary computation methods, such as genetic algorithms [85] or multi-agent systems [86]. In general (but not always), such systems have opportunities for performance creativity. They often have a parameterisation that is simple to change—for example, a fitness function. They also have an emergent [87] output which can sometimes produce unexpected but coherent results.

1.3.7.1 Genetic Programming Jazz Sax

Some of the first researchers to use EC in computer systems for expressive performance were Ramirez and Hazan. They did not start out using EC, beginning with a regression tree system for jazz saxophone [88] (more detail about their machine learning work—applied to violin—can be found in Chap. 5.) This system will be described before moving on to the genetic programming (GP) approach, as it is the basis of their later GP work. A performance decision tree was first built using a learning algorithm called C4.5. This model was built for musicological purposes—to see what kinds of rules were generated—not to generate any performances. The decision tree system had a 3-note-level expressive representation, and music features used to characterise a note included metrical position and some Narmour IR analysis. These features were mapped onto a number of performance actions from the training performances, such as lengthen/shorten note, play note early/late and play note louder/softer. Monophonic audio was used to build this decision tree using the authors' own spectral analysis techniques and five jazz standards at 11 different tempos. The actual performing system was built as a

regression rather than decision tree, thus allowing continuous expressive actions. The continuous performance features simulated were duration, onset and energy variation (i.e. loudness). The learning algorithm used to build the tree was M5Rules, and performances could be generated via MIDI and via audio using the synthesis algorithms. In tests, the resulting correlations with the original performances were 0.72, 0.44 and 0.67 for duration, onset and loudness, respectively. Other modelling methods were tried (linear regression and four different forms of support vector machines) but did not fare as well correlation-wise.

Ramirez and Hazan's next system [58] was also based on regression trees, but these trees were generated using genetic programming (GP), which is ideal for building a population of 'if-then' regression trees. GP was used to search for regression trees that best emulated a set of human audio performance actions. The regression tree models were basically the same as in their previous paper, but in this case a whole series of trees was generated; they were tested for fitness, and then the fittest were used to produce the next generation of trees/programs (with some random mutations added). Fitness was judged based on a distance calculated from a human performance. Creativity and expressive representation are enhanced because, in addition to modelling timing and dynamics, the trees modelled the expressive combining of multiple score notes into a single performance note (consolidation) and the expressive insertion of one or several short notes to anticipate another performance note (ornamentation). These elements are fairly common in jazz saxophone. It was possible to examine these deviations because the fitness function was implemented using an edit distance to measure score edits.

This evolution was continued until average fitness across the population of trees ceased to increase. The use of GP techniques was deliberately applied to give a range of options for the final performance since, as the authors say, 'performance is an inexact phenomenon'. Also, because of the mutation element in genetic programming, there is the possibility of unusual performances being generated. So this CSEMP has quite a good potential for performance creativity. No evaluation was reported of the resulting trees' performances—but average fitness stopped increasing after 20 generations.

1.3.7.2 Sequential Covering Algorithm GAs

The sequential covering algorithm genetic algorithm (GA) by Ramirez and Hazan [59] uses sequential covering to learn performance. Each covering rule is learned using a GA, and a series of such rules are built up, covering the whole problem space. In this paper, the authors return to their first (non-EC) paper's level of expressive representation—looking at note-level deviations without ornamentation or consolidation. However, they make significant improvements over their original non-EC paper. The correlation coefficients for onset, duration and energy/loudness in the original system were 0.72, 0.44 and 0.67—but in this new system, they were 0.75, 0.84 and 0.86—significantly higher. And this system also has the advantage of slightly greater creativity due to its GA approach.

1.3.7.3 Generative Performance GAs

A parallel thread of EC research are Zhang and Miranda's [89] monophonic generative performance GAs which evolve pulse sets [21]. Rather than comparing the generated performances to actual performances, the fitness function here expresses constraints inspired by the generative performance work of Eric Clarke [11]. When a score is presented to the GA system for performance, the system constructs a theoretical timing and dynamics curve for the melody (one advantage of this CSEMP is that this music analysis is automatic). However, this curve is not used directly to generate the actual performance but to influence the evolution. This, together with the GA approach, it increases the performance creativity of the system. The timing curve comes from an algorithm based on Cambouropoulos' [90] local boundary detection model (LBDM)—the inputs to this model are score note timing, pitch and harmonic intervals. The resulting curve is higher for greater boundary strengths. The approximate dynamics curve is calculated from a number of components—the harmonic distance between two notes (based on a measure by Krumhans [91]), the metrical strength of a note (based on the Melisma software [92]) and the pitch height. These values are multiplied for each note to generate a dynamics curve. The expressive representation of this system is the same as the expressive representation of the methodologies used to generate the theoretical curves.

A fitness function is constructed referring to the score representation curves. It has three main elements—fitness is awarded if (1) the pulse set dynamics and timing deviations follow the same direction as the generated dynamics and timing curves; (2) timing deviations are increased at boundaries and (3) timing deviations are not too extreme. Point (1) does not mean that the pulse sets are the same as the dynamics and timing curves, but—all else being equal—that if the dynamics curve moves up between two notes and the pulse set moves up between those two notes, then that pulse set will get a higher fitness than one that moves down there. Regarding point (3), this is reminiscent of the restriction of expression used in the ANN models [38] and the KCCA piano [57] model described earlier. It is designed to preventing the deviations from becoming too extreme.

There has been no formal testing of this GA work, though the authors demonstrate—using an example of part of Schumann's Träumerei and a graphical plot—that the evolved pulse sets are consistent in at least one example with the theoretical timing and dynamics deviation curves. They claim that 'when listening to pieces performed with the evolved pulse sets, we can perceive the expressive dynamics of the piece'. However, more evaluation would be helpful; repeating pulse sets have been shown to not be universally applicable as an expressive action format. Post-Clynes CSEMPs have shown more success using non-cyclic expression.

1.3.7.4 Multi-Agent System with Imitation

Miranda's team [60, 93] developed the above system into a multi-agent system (MAS) with imitation—influenced by Miranda's [94] evolution of music MAS study and inspired by the hypothesis that expressive music performance strategies

emerge through interaction and evolution in the performers' society. It is described in more detail in Chap. 4. In this model, each agent listens to other agents' monophonic performances, evaluates them and learns from those whose performances are better than their own. Every agent's evaluation equation is the same as the fitness function used in the previous GA-based system, and performance deviations are modelled as a hierarchical pulse set. So it has the same expressive representation.

This CSEMP has significant performance creativity, one reason being that the pulse sets generated may have no similarity to the hierarchical constraints of human pulse sets. They are generated mathematically and abstractly from agent imitation performances. So entirely novel pulse set types could be produced by agents that a human would never generate. Another element that contributes to creativity is that although a global evaluation function approach was used, a diversity of performances was found to be produced in the population of agents.

1.3.7.5 Ossia

Like the computational music emotion rule system [31], Dahlstedt's [61] Ossia is a CSEMP 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. Although it is grouped here as an EC learning system, technically Ossia is not a learning system. It is not using EC to learn how to perform like a human but to generate novel compositions and performances. However, it is included in this section because its issues relate more closely to EC and learning systems than to any of the non-learning systems (the same reason applies for the system *pMIMACS* described in the next section). Ossia generates music through a novel representational structure that encompasses both composition and performance—recursive trees (generated by GAs). 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.

This issue of music representation is not something this survey has addressed explicitly, being in itself an issue worthy of its own review, for examples, see [67, 95]. However, a moment will be taken to briefly discuss it now. The representation chosen for a musical system has a significant impact on the functionality—Ossia's representation is what leads to its combined composition and performance generation abilities. The most common music representation mentioned in this chapter

has been MIDI, which is not able to encode musical structure directly. As a result, some MIDI-based CSEMPs have to supply multiple files to the CSEMP, a MIDI file together with files describing musical structure. More flexible representations than MIDI include MusicXML, ENP-score-notation [96], WEDELMUSIC XML [97], MusicXML4R [98] and the proprietary representations used by commercial software such as Sibelius, Finale, Notion and Zenph high-resolution MIDI [99] (which was recently used on a released CD of automated Disklavier re-performances of Glenn Gould).

Many of the performance systems described in this chapter so far transform an expressionless MIDI or audio file into an expressive version. 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 GA to generate a population of trees and judges for fitness using such rules as number of notes per second, repetitivity, amount of silence, pitch variation and level of recursion. These fitness rules were developed heuristically by Dahlstedt through analysis-by-synthesis methods.

Ossia's level of expressive representation is equal to its level of compositional representation. Dahlstedt observes 'The general concept of recapitulation is not possible, as in the common ABA form. This does not matter so much in short compositions, but may be limiting.' So Ossia's expressive representation would seem to be within the A's and B's, giving it a note- to section-level expressive representation. In terms of testing, the system has not been formally evaluated, though it was exhibited as an installation at Gaudeamus Music Week in Amsterdam. Examples are also available on Dahlstedt's website, including a composed suite. The author claims that the sound examples 'show that the Ossia system has the potential to generate and perform piano pieces that could be taken for human contemporary compositions'. The examples on the website are impressive in their natural quality. The question of how to test a combined performance and composition, when that system is not designed to simulate but to create, is a sophisticated problem which will not be addressed here. Certainly, listening tests are a possibility, but these may be biased by the preferences of the listener (e.g. preferring pre-1940s classical music or pop music). Another approach is musicological analysis, but the problem then becomes that musicological tools are not available for all genres and all periods—for example, musicology is more developed for pre-1940 than post-1940 art music.

An example score from Ossia is described which contains detailed dynamics and articulations and subtle tempo fluctuations and rubato. This subtlety raises

another issue—scores generated by Ossia in common music notation had to be simplified to be simply readable by humans. The specification of exact microfeatures in a score can lead to it being unplayable except by computer or the most skilled concert performer. This has a parallel in a compositional movement which emerged in the 1970s, the 'New Complexity', involving composers such as Brian Ferneyhough and Richard Barrett [100]. In the 'New Complexity', elements of the score are often specified down to the microfeature level, and some scores are described as almost unplayable. Compositions such as this, whether by human or computer, bring into question the whole composition/performance dichotomy. (These issues also recall the end of Ian Pace's quote in the first section of this chapter.) However, technical skill limitations and common music notation scores are not necessary for performance if the piece is being written on and performed by a computer. Microfeatures can be generated as part of the computer (or computer-aided) composition process if desired. In systems such as Ossia and CMERS [31], as in the New Complexity, the composition/performance dichotomy starts to break down—the dichotomy is really between macrofeatures and microfeatures of the music.

1.3.7.6 pMIMACS

Before discussing this final system in the survey, another motivation for bringing composition and performance closer in CSEMPs should be highlighted. A significant amount of CSEMP effort is in analysing the musical structure of the score/audio (using methods like that in Chap. 8). However, many computer composition systems generate a piece based on some structure which can often be made explicitly available. So in computer music it is often inefficient to have separate composition and expressive performance systems—i.e. where a score is generated and the 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 CSEMP or—like Ossia—simply combine the systems using, for example, a common representation (where microtiming and microdynamics are seen as an actual part of the composition process). A system which was designed to utilise this combination of performance and composition is pMIMACS, developed by the authors of this survey. It is based on a previous system MIMACS (mimetics-inspired multi-agent composition system), which was developed to solve a specific compositional problem: generating a multi-speaker electronic composition.

pMIMACS combines composition and expressive performance—the aim being to generate contemporary compositions on a computer which, when played back on a computer, do not sound too machine-like. In [60] (Chap. 4), the agents imitate each other's expressive performances of the same piece, whereas in pMIMACS, agents can be performing entirely different pieces of music. 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. 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 MIDI pitch average and standard deviation of their initial tune—this could be interpreted as the range they are comfortable performing in or as the tune they are familiar with performing. 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. 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 a form of performance context, called an excitability state. An excited agent will play its tune with twice the tempo of a non-excited 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 memorise the tune as it hears it, including any performance errors or changes. As the agents perform to each other, they store internally an exponentially growing piece of transforming music. The significant and often smooth deviations in tempo generated by the performance interaction will create a far less robotic-sounding performance than rhythms generated by a quantized palette would do. On the downside, the large-scale rhythmic texture has the potential to become repetitive because of the simplicity of the agents' statistical model of performance skill. Furthermore, the system can generate rests that are so long that the composition effectively comes to a halt for the listener. But overall the MAS is expressing its experience of what it is like to perform the tune, by changing the tempo and dynamics of the tune, and at the same time, this contributes to the composition of the music.

There is also a more subtle form of expression going on relating to the hierarchical structure of the music. The hierarchy develops as agents pass around an ever growing string of phrases. Suppose an agent performs a phrase P and passes it on. Later on, it may receive back a 'super-phrase' containing two other phrases Q and R—in the form QPR. In this case, agent A will perform P faster than Q and R (since it knows P). Now suppose in future A is passed back a super-super-phrase of, say, SQPRTQPRTS, then potentially it will play P fastest, QPR second fastest (since it has played QPR before) and the S and T phrases slowest. So the tempo and amplitude at which an agent performs the parts SQPRTQPRTS are affected by how that phrase was built up hierarchically in the composition/imitation process. Thus, there is an influence on the performance from the hierarchical structure of the

music. This effect is only approximate because of the first-order statistical model of performance skill.

No formal listening tests have been completed yet, but examples of an agent's tune memory after a number of cycles are available from the authors by request. Despite the lack of formal listening tests, pMIMACS is reported here as a system designed from the ground up to combine expressive performance and composition.

1.4   Summary

Before reading this summary, another viewing of Table 1.1 at the start of the chapter may be helpful to the reader. Expressive performance is a complex behaviour with many causative conditions—so it is no surprise that in this chapter almost two thirds of the systems produced have been learning CSEMPs, usually learning to map music features onto expressive actions. Expressive performance actions most commonly included timing and loudness adjustments, with some articulation, and the most common non-custom method for analysis of music features was GTTM, followed by IR. Due to its simplicity in modelling performance, the most common instrument simulated was piano—but interestingly this was followed closely by saxophone—possibly because of the popularity of the instrument in the jazz genre. Despite, and probably because, of its simplicity, MIDI is still the most popular representation.

To help structure the chapter, four primary terms of reference were selected: testing status, expressive representation, polyphonic ability and performance creativity. Having applied these, it can be seen that only a subset of the systems have had any formal testing, and for some of them designing formal tests is a challenge in itself. This is not that unexpected—since testing a creative computer system is an unsolved problem and is discussed more in Chap. 7. Also about half of the systems have only been tested on monophonic tunes. Polyphony and homophony introduce problems both in terms of synchronisation and in terms of music feature analysis (see Chap. 6). Further to music feature analysis, most of the CSEMPs had an expressive representation up to one bar/phrase, and over half did not look at the musical hierarchy. However, avoiding musical hierarchy analysis can have the advantage of increasing automation. We have also seen that most CSEMPs are designed for simulation of human expressive performances, general or specific—a valuable research goal and one which has possibly been influenced by the philosophy of human simulation in machine intelligence research.

The results for the primary terms of reference are summarised in Table 1.5. The numerical measures in columns 1, 2 and 4 are an attempt to quantify observations, scaled from 1 to 10. The more sophisticated the expressive representation of music features (levels of the hierarchy, harmony, etc.), the higher the number in column 1. The more extensive the testing (including informal listening, RenCon

Table 1.5 Summary of the four primary terms of reference

CSEMP	Expressive representation	Testing status	Non-monophonic	Performance creativity
Director Musices	6	10	Y	8
Hierarchical parabola model	8	7		3
Composer pulse	6	9	Y	5
Bach fugue	6	4	Y	3
Rubato	7	4	Y	8
Trumpet synthesis	4	1		3
MIS	6	6	Y	3
ANN piano	4	6		3
Music Plus One	4	7	Y	4
SaxEx	5	1		8
CaRo	3	6		7
Emotional flute	4	6		6
Kagurame	8	8	Y	3
Ha-Hi-Hun	6	6	Y	8
PLCG	4	6		3
Phrase-decomposition/ PLCG	10	6		3
DISTALL	10	9	Y	3
Pop-E	6	10	Y	8
ESP piano	6	6	Y	3
KCCA piano	4	6	Y	3
Drumming	3	6		3
Genetic programming	7	6		6
Sequential covering GAs	6	6		6
Generative performance GAs	8	4		8
MAS with imitation	8	1		9
Ossia	6	4	Y	10
Music emotionality	6	6	Y	10
pMIMACS	8	1		10

submission, formal listening, correlation and/or successful utilisation in the field), the higher the number in column 2. The greater we perceived the potential of a system to enable the creative generation of novel performances, the higher the number in column 4. Obviously, such measures contain some degree of subjectivity but should be a useful indicator for anyone wanting an overview of the field, based on the four elements discussed at the start of this chapter. Figure 1.4 shows a 3-D plot summary of Table 1.5.
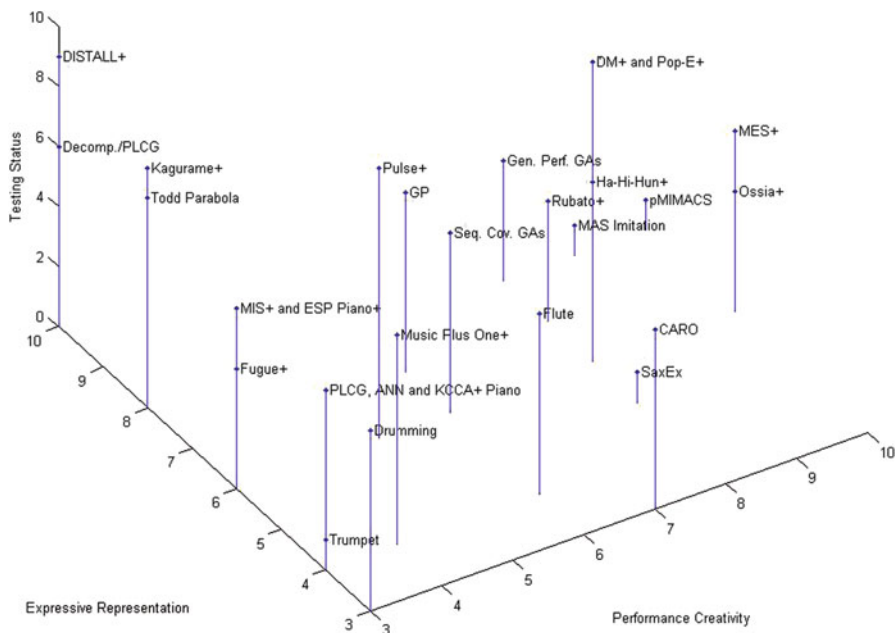
Fig. 1.4 Evaluation of reviewed systems relative to terms of reference ('+' means polyphonic)

1.5   Conclusions

There have been significant achievements in the field of simulating human musical performance in the last 30 years, and there are many opportunities ahead for future improvements in simulation. In fact, one aim of the RenCon competitions is for a computer to win the Chopin competition by 2050. Such an aim begs some philosophical and historical questions but nonetheless captures the level of progress being made in performance simulation. The areas of expressive representation and polyphonic performance appear to be moving forwards. However, the issue of testing and evaluation still requires more work and would be a fruitful area for future CSEMP research.

Another fruitful area for research is around the issue of the automation of the music analysis. Of the eight CSEMPs with the highest expressive representation, almost half of them require some manual input to perform the music analysis. Also, manual marking of the score into phrases is a common requirement. There has been some research into automating musical analysis such as GTTM, as can be seen in Chap. 8. The usage of such techniques, and the investigation of further automation analysis methods specific to expressive performance, would be a useful contribution to the field.

The field could also benefit from a wider understanding of the relationship between performance and composition elements in computer music, for reasons of efficiency,

controllability and creativity. This chapter began by developing four terms of reference which were inspired from research into computer composition systems and has been brought to a close with a pair of systems that question the division between expressive performance and composition. Computer composition and computer performance research can cross-fertilise: performance algorithms for expressing structure and emotion/mood can help composition as well as composition providing more creative and controlled computer performance. Such a cross-fertilisation happens to some degree in Chap. 8. The question is also open as to what forms of non-human expression can be developed and provide whole new vistas of the meaning of the phrase 'expressive performance', perhaps even for human players.

One final observation regards the lack of neurological and physical models of performance simulation. The issue was not included as a part of our terms of reference, since it is hard to objectively quantify. But we would like to address this in closing. Neurological and physical modelling of performance should go beyond ANNs and instrument physical modelling. The human/instrument performance process is a complex dynamical system about which there has been some deeper psychological and physical study. However, attempts to use these hypotheses to develop computer performance systems have been rare. More is being learned about the neural correlates of music and emotion [101–103], and Eric Clarke [104] has written on the importance of physical embeddedness of human performance. But although researchers such as Parncutt [105] (in his virtual pianist approach) and Widmer [106] have highlighted the opportunity for deeper models, there has been relatively little published progress in this area of the CSEMP field, though the issues of physically embedded performance are examined further in Chap. 9.

So to conclude, the overarching focus so far means that there are opportunities for some better tested, more creative and neurological and biomechanical models of human performance. These will be systems which help not only to win the Chopin contest but also to utilise the innate efficiencies and power of computer music techniques. Music psychologists and musicologists will be provided with richer models; composers will be able to work more creatively in the micro-specification domain and more easily and accurately generate expressive performances of their work. And the music industry will be able to expand the power of humanisation tools, creating new efficiencies in recording and performance.

Questions

1. Give two examples of why humans make their performances sound so different to the so-called perfect performance a computer would give.
2. What is the purpose of the 'performance context' module in a generic computer system for expressive music performance?

3. What are two examples of ways in which the performance knowledge system might store its information?
4. Give five reasons that enable computers to perform music expressively.
5. What is the most common form of instrument used in studying computer systems for expressive music performance?
6. What are the two most common forms of expressive performance action?
7. Why is musical structure analysis so significant in computer systems for expressive music performance?
8. In what ways does most Western music usually have a hierarchical structure?
9. What are the potential advantages of combining algorithm composition with expressive performance?
10. Do most of the CSEMPs discussed in this chapter deal with MIDI or audio formats?

References

1. Hiller L, Isaacson L (1959) Experimental music. Composition with an electronic computer. McGraw-Hill, New York
2. Buxton WAS (1977) A composer's introduction to computer music. Interface 6:57–72
3. Roads C (1996) The computer music tutorial. MIT Press, Cambridge
4. Miranda ER (2001) Composing music with computers. Focal Press, Oxford
5. Todd NP (1985) A model of expressive timing in tonal music. Music Percept 3:33–58
6. Friberg A, Sundberg J (1999) Does music performance allude to locomotion? A model of final ritardandi derived from measurements of stopping runners. J Acoust Soc Am 105:1469–1484
7. Pace I (2007) Complexity as imaginative stimulant: issues of rubato, barring, grouping, accentuation and articulation in contemporary music, with examples from Boulez, Carter, Feldman, Kagel, Sciarrino, Finnissy. In: Proceedings of the 5th international Orpheus academy for music, theory, Gent, Belgium, Apr 2007
8. Seashore CE (1938) Psychology of music. McGraw-Hill, New York
9. Palmer C (1997) Music performance. Annu Rev Psychol 48:115–138
10. Gabrielsson A (2003) Music performance research at the millennium. Psychol Music 31:221–272
11. Clarke EF (1998) Generative principles in music performance. In: Sloboda JA (ed) Generative processes in music: the psychology of performance, improvisation, and composition. Clarendon, Oxford, pp 1–26
12. Gabrielsson A, Juslin P (1996) Emotional expression in music performance: between the performer's intention and the listener's experience. Psychol Music 24:68–91
13. Juslin P (2003) Five facets of musical expression: a psychologist's perspective on music performance. Psychol Music 31:273–302
14. Good M (2001) MusicXML for notation and analysis. In: Hewlett WB, Selfridge-Field E (eds) The virtual score: representation, retrieval, restoration. MIT Press, Cambridge, pp 113–124
15. Lerdahl F, Jackendoff R (1938) A generative theory of tonal music. The MIT Press, Cambridge
16. Narmour E (1990) The analysis and cognition of basic melodic structures: the implication-realization model. The University of Chicago Press, Chicago
17. Friberg A, Bresin R, Sundberg J (2006) Overview of the KTH rule system for musical performance. Adv Cognit Psychol 2:145–161

18. Todd NP (1989) A computational model of Rubato. Contemp Music Rev 3:69–88
19. Todd NP (1992) The dynamics of dynamics: a model of musical expression. J Acoust Soc Am 91:3540–3550
20. Todd NP (1995) The kinematics of musical expression. J Acoust Soc Am 97:1940–1949
21. Clynes M (1986) Generative principles of musical thought: integration of microstructure with structure. Commun Cognit 3:185–223
22. Clynes M (1995) Microstructural musical linguistics: composer's pulses are liked best by the musicians. Cognit: Int J Cognit Sci 55:269–310
23. Johnson ML (1991) Toward an expert system for expressive musical performance. Computer 24:30–34
24. Dannenberg RB, Derenyi I (1998) Combining instrument and performance models for high-quality music synthesis. J New Music Res 27:211–238
25. Dannenberg RB, Pellerin H, Derenyi I (1998) A study of trumpet envelopes. In: Proceedings of the 1998 international computer music conference, Ann Arbor, Michigan, October 1998. International Computer Music Association, San Francisco, pp 57–61
26. Mazzola G, Zahorka O (1994) Tempo curves revisited: hierarchies of performance fields. Comput Music J 18(1):40–52
27. Mazzola G (2002) The topos of music – geometric logic of concepts, theory, and performance. Birkhäuser, Basel/Boston
28. Hashida M, Nagata N, Katayose H (2006) Pop-E: a performance rendering system for the ensemble music that considered group expression. In: Baroni M, Addessi R, Caterina R, Costa M (eds) Proceedings of 9th international conference on music perception and cognition, Bologna, Spain, August 2006. ICMPC, pp 526–534
29. Sethares W (2004) Tuning, timbre, spectrum, scale. Springer, London
30. Finn B (2007) Personal communication
31. Livingstone SR, Muhlberger R, Brown AR, Loch A (2007) Controlling musical emotionality: an affective computational architecture for influencing musical emotions. Digit Creat 18:43–53
32. Katayose H, Fukuoka T, Takami K, Inokuchi S (1990) Expression extraction in virtuoso music performances. In: Proceedings of the 10th international conference on pattern recognition, Atlantic City, New Jersey, USA, June 1990. IEEE Press, Los Alamitos, pp 780–784
33. Aono Y, Katayose H, Inokuchi S (1997) Extraction of expression parameters with multiple regression analysis. J Inf Process Soc Jpn 38:1473–1481
34. Ishikawa O, Aono Y, Katayose H, Inokuchi S (2000) Extraction of musical performance rule using a modified algorithm of multiple regression analysis. In: Proceedings of the international computer music conference, Berlin, Germany, August 2000. International Computer Music Association, San Francisco, pp 348–351
35. Canazza S, Drioli C, De Poli G, Roda A, Vidolin A (2000) Audio morphing different expressive intentions for multimedia systems. IEEE Multimed 7:79–83
36. Canazza S, De Poli G, Drioli C, Roda A, Vidolin A (2001) Expressive morphing for interactive performance of musical scores. In: Proceedings of first international conference on WEB delivering of music, Florence, Italy, Nov 2001. IEEE, Los Alamitos, pp 116–122
37. Canazza S, De Poli G, Roda A, Vidolin A (2003) An abstract control space for communication of sensory expressive intentions in music performance. J New Music Res 32:281–294
38. Bresin R (1998) Artificial neural networks based models for automatic performance of musical scores. J New Music Res 27:239–270
39. Camurri A, Dillon R, Saron A (2000) An experiment on analysis and synthesis of musical expressivity. In: Proceedings of 13th colloquium on musical informatics, L'Aquila, Italy, Sept 2000
40. Arcos JL, De Mantaras RL, Serra X (1997) SaxEx: a case-based reasoning system for generating expressive musical performances. In: Cook PR (eds) Proceedings of 1997 international computer music conference, Thessalonikia, Greece, Sept 1997. ICMA, San Francisco, pp 329–336
41. Arcos JL, Lopez De Mantaras R, Serra X (1998) Saxex: a case-based reasoning system for generating expressive musical performance. J New Music Res 27:194–210

42. Arcos JL, Lopez De Mantaras R (2001) An interactive case-based reasoning approach for generating expressive music. J Appl Intell 14:115–129
43. Suzuki T, Tokunaga T, Tanaka H (1999) A case based approach to the generation of musical expression. In: Proceedings of the 16th international joint conference on artificial intelligence, Stockholm, Sweden, Aug 1999. Morgan Kaufmann, San Francisco, pp 642–648
44. Suzuki T (2003) Kagurame phase-II. In: Gottlob G, Walsh T (eds) Proceedings of 2003 international joint conference on artificial intelligence (Working Notes of RenCon Workshop), Acapulco, Mexico, Aug 2003. Morgan Kauffman, Los Altos
45. Hirata K, Hiraga R (2002) Ha-Hi-Hun: performance rendering system of high controllability. In: Proceedings of the ICAD 2002 RenCon workshop on performance rendering systems, Kyoto, Japan, July 2002, pp 40–46
46. Widmer G (2000) Large-scale induction of expressive performance rules: first quantitative results. In: Zannos I (eds) Proceedings of the 2000 international computer music conference, Berlin, Germany, Sept 2000. International Computer Music Association, San Francisco, 344–347
47. Widmer G (2002) Machine discoveries: a few simple, robust local expression principles. J New Music Res 31:37–50
48. Widmer G (2003) Discovering simple rules in complex data: a meta-learning algorithm and some surprising musical discoveries. Artif Intell 146:129–148
49. Widmer G, Tobudic A (2003) Playing Mozart by analogy: learning multi-level timing and dynamics strategies. J New Music Res 32:259–268
50. Tobudic A, Widmer G (2003) Relational ibl in music with a new structural similarity measure. In: Horvath T, Yamamoto A (eds) Proceedings of the 13th international conference on inductive logic programming, Szeged, Hungary, Sept 2003. Springer Verlag, Berlin, pp 365–382
51. Tobudic A, Widmer G (2003) Learning to play Mozart: recent improvements. In: Hirata K (eds) Proceedings of the IJCAI'03 workshop on methods for automatic music performance and their applications in a public rendering contest (RenCon), Acapulco, Mexico, Aug 2003
52. Raphael C (2001) Can the computer learn to play music expressively? In: Jaakkola T, Richardson T (eds) Proceedings of eighth international workshop on artificial intelligence and statistics, 2001. Morgan Kaufmann, San Francisco, pp 113–120
53. Raphael C (2001) A Bayesian network for real-time musical accompaniment. Neural Inf Process Sys 14:1433–1440
54. Raphael C (2003) Orchestra in a box: a system for real-time musical accompaniment. In: Gottlob G, Walsh T (eds) Proceedings of 2003 international joint conference on artificial intelligence (Working Notes of RenCon Workshop), Acapulco, Mexico, Aug 2003. Morgan Kaufmann, San Francisco, pp 5–10
55. Grindlay GC (2005) Modelling expressive musical performance with Hidden Markov Models. PhD thesis, University of Santa Cruz, CA
56. Carlson L, Nordmark A Wikilander R (2003) Reason version 2.5 – getting started. Propellerhead Software
57. Dorard L, Hardoon DR, Shawe-Taylor J (2007) Can style be learned? A machine learning approach towards 'performing' as famous pianists. In: Music, brain and cognition workshop, NIPS 2007, Whistler, Canada
58. Hazan A, Ramirez R (2006) Modelling expressive performance using consistent evolutionary regression trees. In: Brewka G, Coradeschi S, Perini A, Traverso P (eds) Proceedings of 17th European conference on artificial intelligence (Workshop on Evolutionary Computation), Riva del Garda, Italy, Aug 2006. IOS Press, Washington, DC
59. Ramirez R, Hazan A (2007) Inducing a generative expressive performance model using a sequential-covering genetic algorithm. In: Proceedings of 2007 genetic and evolutionary computation conference, London, UK, July 2007. ACM Press, New York
60. Miranda ER, Kirke A, Zhang Q (2010) Artificial evolution of expressive performance of music: an imitative multi-agent systems approach. Comput Music J 34(1):80–96
61. Dahlstedt P (2007) Autonomous evolution of complete piano pieces and performances. In: Proceedings of ECAL 2007 workshop on music and artificial life (MusicAL 2007), Lisbon, Portugal, Sept 2007

62. Papadopoulos G, Wiggins GA (1999) AI methods for algorithmic composition: a survey, a critical view, and future prospects. In: Proceedings of the AISB'99 symposium on musical creativity. AISB, Edinburgh
63. Hiraga R, Bresin R, Hirata K, RenCon KH (2004) Turing test for musical expression proceedings of international conference on new interfaces for musical expression. In: Nagashima Y, Lyons M (eds) Proceedings of 2004 new interfaces for musical expression conference, Hamatsu, Japan, June 2004. Shizuoka University of Art and Culture, ACM Press, New York pp 120–123
64. Arcos JL, De Mantaras RL (2001) The SaxEx system for expressive music synthesis: a progress report. In: Lomeli C, Loureiro R (eds) Proceedings of the workshop on current research directions in computer music, Barcelona, Spain, Nov 2001. Pompeu Fabra University, Barcelona, pp 17–22
65. Church M (2004) The mystery of Glenn Gould. Independent Newspaper, Published by Independent Print Ltd, London, UK
66. Kirke A, Miranda ER (2007) Capturing the aesthetic: radial mappings for cellular automata music. J ITC Sangeet Res Acad 21:15–23
67. Anders T (2007) Composing music by composing rules: design and usage of a generic music constraint system. PhD thesis, University of Belfast
68. Tobudic A, Widmer G (2006) Relational IBL in classical music. Mach Learn 64:5–24
69. Sundberg J, Askenfelt A, Fryden L (1983) Musical performance. A synthesis-by-rule approach. Comput Music J 7:37–43
70. Bresin R, Friberg A (2000) Emotional coloring of computer-controlled music performances. Comput Music J 24:44–63
71. Friberg A (2006) pDM: an expressive sequencer with real-time control of the KTH music-performance rules. Comput Music J 30:37–48
72. Desain P, Honing H (1993) Tempo curves considered harmful. Contemp Music Rev 7:123–138
73. Thompson WF (1989) Composer-specific aspects of musical performance: an evaluation of Clynes's theory of pulse for performances of Mozart and Beethoven. Music Percept 7:15–42
74. Repp BH (1990) Composer's pulses: science or art. Music Percept 7:423–434
75. Hashida M, Nagata N, Katayose H (2007) jPop-E: an assistant system for performance rendering of ensemble music. In: Crawford L (eds) Proceedings of 2007 conference on new interfaces for musical expression (NIME07), New York, NY, pp 313–316
76. Meyer LB (1957) Meaning in music and information theory. J Aesthet Art Crit 15:412–424
77. Canazza S, De Poli G, Drioli C, Roda A, Vidolin A (2004) Modeling and control of expressiveness in music performance. Proc IEEE 92:686–701
78. De Poli G (2004) Methodologies for expressiveness modeling of and for music performance. J New Music Res 33:189–202
79. Lopez De Mantaras R, Arcos JL (2002) AI and music: from composition to expressive performances. AI Mag 23:43–57
80. Mitchell T (1997) Machine learning. McGraw-Hill, New York
81. Emde W, Wettschereck D (1996) Relational instance based learning. In: Saitta L (eds) Proceedings of 13th international conference on machine learning, Bari, Italy, July 1996. Morgan Kaufmann, San Francisco, pp 122–130
82. Wright M, Berdahl E (2006) Towards machine learning of expressive microtiming in Brazilian drumming. In: Zannos I (eds) Proceedings of the 2006 international computer music conference, New Orleans, USA, Nov 2006. ICMA, San Francisco, pp 572–575
83. Dixon S, Goebl W, Widmer G (2002) The performance worm: real time visualisation of expression based on Langrer's tempo-loudness animation. In: Proceedings of the international computer music conference, Goteborg, Sweden, Sept, pp 361–364
84. Sholkopf B, Smola A, Muller K (1998) Nonlinear component analysis as a kernel eigenvalue problem, Neural computation 10. MIT Press, Cambridge, MA, pp 1299–1319
85. Mitchell M (1998) Introduction to genetic algorithms. The MIT Press, Cambridge
86. Kirke A (1997) Learning and co-operation in mobile multi-robot systems. PhD thesis, University of Plymouth

87. Chalmers D (2006) Strong and weak emergence. In: Clayton P, Davies P (eds) The re-emergence of emergence. Oxford University Press, Oxford
88. Ramirez R, Hazan A (2005) Modeling expressive performance in Jazz. In: Proceedings of 18th international Florida Artificial Intelligence Research Society conference (AI in Music and Art), Clearwater Beach, FL, USA, May 2005. AAAI Press, Menlo Park, pp 86–91
89. Zhang Q, Miranda ER (2006) Towards an interaction and evolution model of expressive music performance. In: Chen Y, Abraham A (eds) Proceedings of the 6th international conference on intelligent systems design and applications, Jinan, China, Oct 2006. IEEE Computer Society, Washington, DC, pp 1189–1194
90. Cambouropoulos E (2001) The local boundary detection model (LBDM) and its application in the study of expressive timing. In: Schloss R, Dannenberg R (eds) Proceedings of the 2001 international computer music conference, Havana, Cuba, Sept 2001. International Computer Music Association, San Francisco
91. Krumhansl C (1991) Cognitive foundations of musical pitch. Oxford University Press, Oxford
92. Temperley D, Sleator D (1999) Modeling meter and harmony: a preference rule approach. Comput Music J 23:10–27
93. Zhang Q, Miranda ER (2007) Evolving expressive music performance through interaction of artificial agent performers. In: Proceedings of ECAL 2007 workshop on music and artificial life (MusicAL 2007), Lisbon, Portugal, Sept
94. Miranda ER (2002) Emergent sound repertoires in virtual societies. Comput Music J 26(2):77–90
95. Dannenberg RB (1993) A brief survey of music representation issues, techniques, and systems. Comput Music J 17:20–30
96. Laurson M, Kuuskankare M (2003) From RTM-notation to ENP-score-notation. In: Proceedings of Journées d'Informatique Musicale 2003, Montbéliard, France
97. Bellini P, Nesi P (2001) WEDELMUSIC format: an XML music notation format for emerging applications. In: Proceedings of first international conference on web delivering of music, Florence, Nov 2001. IEEE Press, Los Alamitos, pp 79–86
98. Good M (2006) MusicXML in commercial applications. In: Hewlett WB, Selfridge-Field E (eds) Music analysis east and west. MIT Press, Cambridge, MA, pp 9–20
99. Atkinson JJS (2007) Bach: the Goldberg variations. Stereophile, Sept 2007
100. Toop R (1988) Four facets of the new complexity. Contact 32:4–50
101. Koelsch S, Siebel WA (2005) Towards a neural basis of music perception. Trends Cogn Sci 9:579–584
102. Britton JC, Phan KL, Taylor SF, Welsch RC, Berridge KC, Liberzon I (2006) Neural correlates of social and nonsocial emotions: an fMRI study. Neuroimage 31:397–409
103. Durrant S, Miranda ER, Hardoon D, Shawe-Taylor J, Brechmann A, Scheich H (2007) Neural correlates of tonality in music. In: Proceedings of music, brain, cognition workshop – NIPS Conference, Whistler, Canada
104. Clarke EF (1993) Generativity, mimesis and the human body in music performance. Contemp Music Rev 9:207–219
105. Parncutt R (1997) Modeling piano performance: physics and cognition of a virtual pianist. In: Cook PR (eds) Proceedings of 1997 international computer music conference, Thessalonikia, Greece, Sept 1997. ICMA, San Francisco, pp 15–18
106. Widmer G, Goebl W (2004) Computational models of expressive music performance: the state of the art. J New Music Res 33:203–216