An overview of computer systems for expressive music performance

Article · December 2009

DOI: 10.1007/978-1-4471-4123-5_1

Alexis Kirke
University of Plymouth

Eduardo Reck Miranda
University of Plymouth

A Survey of Computer Systems for Expressive Music Performance

ALEXIS KIRKE and EDUARDO RECK MIRANDA

University of Plymouth

We present a survey of research into automated and semiautomated computer systems for expressive performance of music. We will examine the motivation for such systems and then examine the majority of the systems developed over the last 25 years. To highlight some of the possible future directions for new research, the review uses primary terms of reference based on four elements: testing status, expressive representation, polyphonic ability, and performance creativity.

Categories and Subject Descriptors: J.5 [**Arts and Humanities**]: *Performing arts (e.g. dance, music)*

General Terms: Algorithms, Experimentation, Human Factors

Additional Key Words and Phrases: Music performance, generative performance, computer music, machine learning

ACM Reference Format:

Kirke, A. and Miranda, E. R. 2009. A survey of computer systems for expressive music performance. ACM Comput. Surv. 42, 1, Article 3 (December 2009), 41 pages.
DOI = 10.1145/1592451.1592454, http://doi.acm.org/10.1145/1592451.1592454

1. INTRODUCTION

In the early 1980s the seeds of a problem were sown as a result of synthesizers being developed and sold with built-in sequencers. The introduction of MIDI into this equation led to an explosion in the use of sequencers and computers, thanks to the new potential for connection and synchronization. These computers and sequencers performed their stored tunes in perfect metronomic time, a performance which sounded robotic. They sounded robotic because human performers normally perform expressively—for example, speeding up and slowing down while playing, and changing how loudly they play. The performers' changes in tempo and dynamics allow them to express a fixed score—hence the term *expressive* performance [Widmer and Goebl 2004]. However, rather than looking for ways to give the music performances more humanlike expression, pop performers developed new types of music, such as synth-pop and dance music, that actually utilized this metronomic perfection to generate robotic performances.

This work has been funded by the U.K. EPSRC project "Learning the Structure of Music" (EPD063612-1).

Authors' addresses: Interdisciplinary Center for Computer Music Research (ICCMR), University of Plymouth, Drake Circus, Plymouth, Devon PL4 8AA, U.K.; email: {Alexis.Kirke,Eduardo.Miranda}@Plymouth.ac.uk.

DOI 10.1145/1592451.1592454 http://doi.acm.org/10.1145/1592451.1592454

Outside of pop, the uptake of sequencers for performance (as opposed to for composition) was less enthusiastic, and except for occasional novelties like *Snowflakes Are Dancing* by Tomita [Kajanová 2002], computer performance of classical music was a rarity. 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 [Hiller and Isaacson 1959]. Since then there has been a large body of such music and research published, with many successful systems produced for automated and semiautomated computer composition [Buxton 1977; Roads 1996; Miranda 2001]. 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 among pop performers, and up to 1987—when Yamaha released its first Disklavier MIDI piano—there were only two or three researchers publishing on algorithms for the expressive performance of music [Todd 1985; Sundberg et al. 1983]. However, from the end of the 1980s on there was an increasing interest in automated and semiautomated computer systems for expressive music performance (CSEMPs). 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, and 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.

In this article we will review the majority of research on automated and semiautomated CSEMPs. By *automated*, we refer to the ability of the system—once setup 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 setup, but then can be presented with multiple pieces which will be played autonomously. A semiautomated system is one which requires some manual input from the user (for example, a musicological analysis) to deal with a new piece.

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 article 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 he or she approaches the end of the piece [Friberg and Sundberg 1999]. 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 has offered up the following as a familiar historical model for the development of notation (though he suggests that overall it constitutes an oversimplification) [Pace 2009, pp. 151–152]:

> In the Middle Ages and to a lesser extent 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-20th-century music there was a tradition of performers making additions to a performance which were not marked in the score (though the reason Pace called 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 on this pre-20th-century (specifically Baroque, Classical, and Romantic) music performance. The earliest studies began with Seashore [1938], and good overviews include Palmer [1997] and Gabrielsson [2003]. 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 [Palmer 1997]. 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, 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. 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 down being correlated to the importance of the boundary [Clarke 1998]. 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 [Gabrielsson and Juslin 1996]. For a discussion of other factors involved in human expressive performance, we refer the reader to Juslin [2003].

1.2. Computer Expressive Performance

Having examined human expressive performance, the question now becomes why we should 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 [Seashore 1938; Palmer 1997; Gabrielsson 2003]. 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 modeling, the model itself can give the researcher greater insight into the mechanisms inherent in that which is being modeled.

(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 the 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 (for example, in a computer game which generates mood music based on what is happening in the game).

(4) *Playing data files.* A large number of nonexpressive data files in formats like MIDI and MusicXML [Good 2001] 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 practice by playing along with recordings with their solo parts 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.

2. A GENERIC FRAMEWORK FOR PREVIOUS RESEARCH IN COMPUTER EXPRESSIVE PERFORMANCE

Figure 1 shows a generic model for the framework that most (but not all) previous research into automated and semiautomated CSEMPs tends to have followed. The modules of this diagram are described in the main text.

—*Performance knowledge.* This is the core of any performance system. It is the set of rules or associations that controls 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 equations 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.

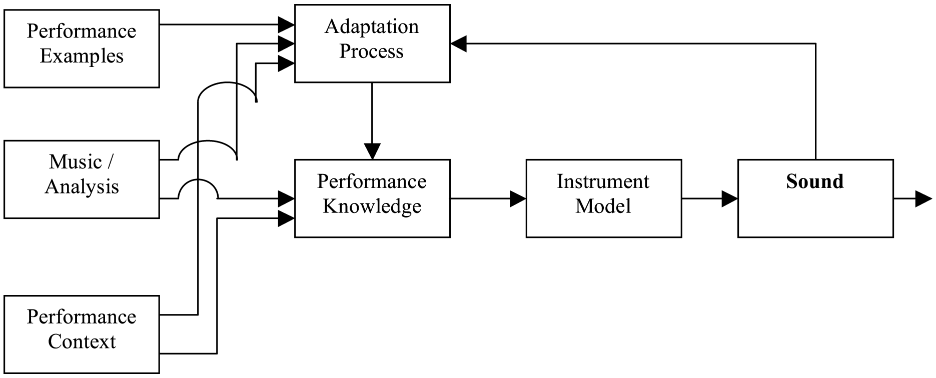

Fig. 1. Generic model for most current CSEMPs.

—*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 forms) 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 [1983] Generative Theory of Tonal Music, Narmour's [1990] Implication Realization, and various bespoken musical measurements. The analysis may be automated, manual, or a combination of the two.

—*Performance context*. Another element which will effect 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 the actual musical output of the performance system and decides if it is appropriate. If not, the performance knowledge can be adjusted to try to improve the musical output performance. This is the reason that in Figure 1 there is a line going from the sound module back to the adaptation procedure module. The adaptation procedure 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 is music/analysis and performance examples.

—*Performance examples*. One important element that can be incorporated into the performance knowledge building is the experience of past human performances. These examples can be used by the adaptation procedure to analyze 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 the 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. However, a few simulation systems use nonkeyboard instruments, for example, saxophone and trumpet. In these cases the issue of a performance is more than just expressiveness. Just simulating a humanlike performance, even if it is nonexpressive, on these instruments is nontrivial. So systems simulating expressive performance on such instruments may require a relatively complex instrument model in addition to expressive performance elements.

2.1. Modules of Systems Reviewed

Table I lists the systems reviewed and the sections they are reviewed in, together with information about their modules. This information will be explained in detail in the main text. Note that 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 I and throughout the article. Table II lists these abbreviations and their meaning.

Before discussing the primary terms of reference for this review, 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. Papadopoulos and Wiggins [1999] discussed the evaluation issue in a different but closely related area—computer algorithmic composition systems. They listed four points that they saw 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 article.
(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 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 review. The three primary terms of reference 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.

2.2. Primary Terms of Reference for Systems Reviewed

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. There are three main approaches to CSEMP testing: (a) trying to simulate a particular human performance or an average of human

Table I. Systems Reviewed

CSEMP	Performance Knowledge	Music	Analysis	Performance Context	Adaptation Process	Performance Examples	Instrument Model (or Application)	Performance Actions
Director Musices (3.1.1)	Rules	MIDI, score	Custom	Mood space	—	MIDI performances	All	T/D/A/P
Hierarchical Parabola (3.1.2)	Parabola equation	MIDI, score	GTTM TSR	—	—	—	Piano	T/D
Composer Pulse (3.1.3)	Multiplier set	MIDI	—	—	Manual	Tapping	All	T/D
Bach Fugue (3.1.4)	Rules	Score	Custom	—	Manual	Books, experts	Keyboard	T/A
Trumpet Synthesis (3.1.5)	Linear model	Audio, score	Custom	—	Manual	Audio performances	Trumpet	A/P/O
Rubato (3.1.6)	Operators	MIDI, score	Custom	—	Manual	—	All	T /D
Pop-E (3.1.7)	Rule based	MIDI, score	GTTM, Custom	—	—	MIDI performances	Piano	T/D
Hermode Tuning (3.1.8)	Rule based	MIDI	Custom	—	—	—	All	P
Sibelius (3.1.9)	Rule based	Score	Custom	—	—	Unknown	—	Unknown
CMERS (3.1.10)	Linear model	MIDI, score	Custom	Mood space	—	Performances and experiments	Piano	T/D/P/O
MIS (3.2.1)	Linear model	MIDI, score	GTTM/Meyer	—	Regression with ANDs	Audio performances	Piano	T/D/A
CaRo (3.2.2)	Linear model	Audio	Custom	Mood space	PCA, linear regression	Audio performances by mood	All	T/K/D/A
ANN Piano (3.3.1)	ANN	MIDI	Custom	—	ANN training	MIDI performances	Piano	T/D
Emotional Flute (3.3.2)	ANN and rules	Audio, score	Custom	Mood space	ANN training	Audio performances by mood	Flute	T/D/V
SaxEx (3.4.1)	Fuzzy Rules	Audio, score	Narmour/IR/ GTTM, custom	Mood space	CBR training	Audio performances by mood	Saxophone	T/D/V/K
Kagurame (3.4.2)	Rules	MIDI, score	Custom	Performance conditions	CBR training	MIDI performances by context	Piano	T/D/A
Ha-Hi-Hun (3.4.3)	Rules	MIDI, score	GTTM TSR	Language performance conditions	CBR training	MIDI performances by condition	Piano	T/D

(*continued on next page*)

Table I. (*Continued*)

CSEMP	Performance Knowledge	Music	Analysis	Performance Context	Adaptation Process	Performance Examples	Instrument Model (or Application)	Performance Actions
PLCG (3.4.4)	Learned rules	MIDI, score	Custom	—	Meta- sequential learning	MIDI performances	Piano	T/D/A
PLCG/Phrase-decomposition (3.4.5)	Learned rules	MIDI, score	Custom, harmonic by musicologist	—	CBR training, Meta- sequential learning	MIDI performances	Piano	T/D/A
DISTALL (3.4.6)	CBR	MIDI, score	Custom, harmonic by musicologist	—	CBR training	MIDI performances	Piano	T/D/A
Music Plus One (3.5.1)	BBN	Audio/MIDI, score	Custom	Soloist tempo	BBN training	Audio soloist performances	All	T/D
ESP Piano (3.5.2)	HMM	MIDI, score	Custom	—	HMM training	MIDI performances	Piano	T/D/A
Drumming (3.6.1)	Nonlinear mapping	Audio	Custom	—	KRR, GPR, kNN	Audio performances	Drums	T
KCCA Piano (3.6.2)	Nonlinear mapping	Worm, score	Custom	—	KCCA	Performance worm	Piano	T/D
Genetic Programming (3.7.1)	Regression trees	Audio	IR, custom	—	Genetic programming	Audio performances	Saxophone	T/D/A/N
Sequential Covering GAs (3.7.2)	Rule based	Audio	IR, custom	—	Sequential covering by GA	Audio performances	Saxophone	T/D/A
Generative Performance GAs (3.7.3)	Pulse set	MIDI, score	LBDM, Kruhmans, Melisma	—	GA	None	Piano	T/D
MAS with Imitation (3.7.4)	Pulse set	MIDI, score	LBDM, Kruhmans, Melisma	—	Imitation	None	Piano	T/D
Ossia (3.7.5)	Fitness rules	MIDI	—	—	—	None	Piano	T/D/P
pMIMACS (3.7.6)	Rule based (2)	MIDI	Performance skill	Excitability state	Imitation	None	Piano	T/D

Table II. Abbreviations Used in Table I and Throughout Article

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/Realization Theory of Melody
K	Attack
KCCA	Kernel canonical correlation analysis
kNN	*k*-Nearest neighbor
KRR	Kernel ridge regression
LBDM	Local Boundary Detection Model of Cambouropoulos
MAS	Multiagent system
CMERS	Computational Music Emotion Rule System by Livingstone et al.
MIDI	Musical Intrument Digital Interface
MIMACS	Memetics-inspired multiagent composition system
MusicXML	Music Extensible Markup Language
MIS	Music Intepretation 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

performances, (b) trying to create a performance which does not sound machinelike, 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 nonexperts. 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, which has been described as a "musical Turing Test," called the *RenCon (Contest for Performance Rendering Systems)* Workshop, has been held [Hiraga et al. 2004]. About a third of the systems we will review have been entered into a RenCon competition—see Table III 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 III 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 conference. Scores are given for "humanness," and "expressiveness," giving an overall "preference" score. It is the preference score we will focus on in the review. The size of RenCon juries and their criteria have varied over time. In

Table III. RenCon Placings of CSEMPS in This Article

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)

previous 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 were not the ideal evaluation for automated and semiautomated CSEMPs. However, they are the only published common forum available, so in the spirit of points (1) and (2) from Papadopoulos and Wiggins [1999], they will be referred to where possible in our review.

From 2008 on, the competition has had three sections: an “autonomous” section, a “type-in” section, and the 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 were presented with two 1-min pieces of unseen music: one in the style of Chopin and one in the style of Mozart. An award was 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 nonmonophonic piece of music. Monophonic music has only one note playing at a time, whereas nonmonophonic 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 [Arcos and de Mantaras 2001] focuses on a single monophonic instrument—the saxophone. However, as will be seen, there are also systems like the ESP Piano System [Grindlay 2005] and Pop-E [Hashida et al. 2006] that are designed explicitly to work with nonmonophonic 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 nonmonophonic performance, could lead to parts having their own expressive timing deviations. This would cause an unsynchronised and unattractive performance [Friberg et al. 2006]. 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.

The third term of reference in this review, *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 [Bresin 1998] was designed to simulate human performances (an important research goal), but not to create novel performances, whereas a system like Director Musices

[Friberg et al. 2006], although also designed to capture human performance strategies, has a parameterization ability which can be creatively manipulated to generate entirely novel performances. 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 realized 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 that has only three basic performances 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 review we are only concerned with automated and semiautomated 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 a creative and novel performance is often applauded in human performers. For example Glenn Gould created highly novel expressive performances of pieces of music and has been described as having a vivid musical imagination [Church 2004]. 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 that proposed by Kirke and Miranda [2007a], whereas others may be able to plan at the higher structure level, generating forms like ABCBA, for example, Anders [2007]. 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 analyzed 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 analyzed—as is done in Widmer's DISTALL system [Widmer 2003; Widmer and Tobudic 2003]. The greater the expressive representation of a CSEMP, the more of the music features it can potentially express.

So to summarize, our four primary terms of reference are as follows:

—testing status,

—expressive representation,

—polyphonic ability, and

—performance creativity.

At some point in the description of each system, these points will be implicitly or explicitly addressed, and are summarized at the end of the article (see Table V). 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 which will help to show potential directions for future research. What now follows is the actual descriptions of the CSEMPs, divided into a number of groups.

3. A SURVEY OF COMPUTER SYSTEMS FOR EXPRESSIVE MUSIC PERFORMANCE

The review presented here is meant to be representative rather than exhaustive but will cover the majority of published automated and semiautomated CSEMP systems to date. Each CSEMP is grouped according to how their performance knowledge is built—that is, by learning method. This provides a manageable division of the field, shows which learning methods are most popular, and shows where there is room for development in the building of performance knowledge models. The grouping will be as follows:

(1) nonlearning (10 systems),
(2) linear regression (two systems),
(3) artificial neural networks (two systems),
(4) rule/case-based learning (six systems),
(5) statistical graphical models (two systems),
(6) other regression methods (two systems), and
(7) evolutionary computation (six 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 nonlearning, 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.

3.1. Nonlearning Systems

3.1.1. Director Musices. Director Musices (DM) [Sundberg et al. 1983; Friberg et al. 2006] has been an ongoing project since 1982. Researchers including violinist Lars Fryden 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, the higher the pitch, the louder the note is played, or during an upward run of notes, the piece is played faster. Another DM rule is *phrase arch*, which defines a "rainbow" shape curve of tempo and dynamics over a note grouping, such as a phrase. The performance speeds up and gets louder toward the center of a note grouping and tails off again in tempo and dynamics toward the end of the grouping. The curves are generated at each level of the hierarchy, from the phrase level upward (Figure 2) and added to get the performance. This rule gives DM a higher expressive representation.

Each rule's equation has a numeric '*k-value*'—the higher the k-value the more effect the rule will have, and a k-value of zero switches the rule off. The results of the equations are added together linearly to get the final performance. Thanks to the adjustable

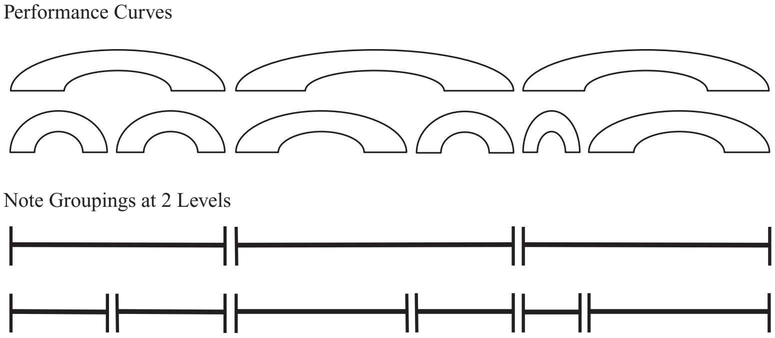

Fig. 2. Applying phrase arch tempo curves to multiple levels of the hierarchy.

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 semiautomated 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). However some manual score analysis is required—for example, harmonic analysis and marking up of note group starts and ends.

Rules are also included for dealing with nonmonophonic music [Friberg et al. 2006]. The *Melodic-sync* rule generates a new voice consisting of all timings in all other voices (if two voices have simultaneous notes, Then the note with the greatest melodic tension is selected). Then all rules are applied to this synchronization voice, and resulting durations are mapped back onto the original voices. The *bar-sync* rule can also be applied to make all voices resynchronize at each bar end.

DM is also able to deal with some performance contexts, specifically emotional expression [Bresin and Friberg 2000], drawing on work by Gabrielsson and Juslin [1996]. Listening experiments were used to define the k-value settings on the DM rules for expressing emotions. The music used was a Swedish nursery rhyme and a computer-generated piece in a minor mode written using Cope's [1992] algorithmic composition system in the musical style of Chopin. Six rules were used from DM to generate multiple performances of each piece. Subjects were asked to identify a performance emotion from the list of 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. For example, for "tenderness" interonset interval was lengthened by 30%, sound level reduced by 6 dB, and two other rules were used: the final ritardando rule (slowing down at the end of a piece) and the duration contrast rule (if two adjacent notes have contrasting durations, increase this contrast).

The most recent development in Director Musices has been the real-time generation of performances using a version of the system called pDM [Friberg 2006]. pDM essentially acts as an expressive sequencer, allowing the adjustment of rule parameters during playback and the live incorporation of the changes as the playback continues. Unlike pDM, many CSEMPs in this survey receive the inputs and parameters and the whole piece of music, process the data, and, when this processing is complete, generate a performance 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 Friberg [1995], k-values were adjusted by a search algorithm, based on 28 human performances of nine bars of of Schumann's *Träumerei*. A good correlation was found between the human performances and the resulting DM performance. Another experiment involved manually fitting to one human performance the first 20 bars of the Adagio in Mozart's sonata K.332 Sundberg et al. [2003]. The correlations were found to be low, unless the k-values were allowed to change dynamically when the piece was performed. An attempt was made to fit k-values to a larger corpus of piano music using genetic algorithms [Kroiss 2000], and the results were found to give a low correlation as well. In an attempt to overcome this, Zanon and de Poli [2003] allowed k-values to vary in a controlled way over a piece of music. This was tested on Beethoven's *Sonatine in G Major* and Mozart's K.332 piano sonata (the slow movement)—but the results were found to be poor for the Beethoven. In the first RenCon in 2002, the second prize went to a DM-generated performance; however, the first-place system (a manually generated performance) was voted for by 80% of the jury. In RenCon 2005, a Director Musices default-settings (i.e., automated) performance of Mozart's *Minuette KV 1(1e)* came in a very close second in the competition, behind Pop-E (Section 3.1.7). 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.

3.1.2. Hierarchical Parabola Model. Todd's [1985; 1989; 1992; 1995] Hierarchical Parabola Model is arguably 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, this model uses a parabolic shape, similar to DM's phrase arch rainbow (though the Hierarchical Parabola Model pre-dates the hierarchical phrase arch rule). This parabolic shape is consistent with Newtonian kinematics. For loudness, the model uses a "the faster the louder" rule, creating a dynamics curve as well.

A key difference between DM's phrase arch and this model is that the Hierarchical Parabola Model analyzes multiple levels of the hierarchy using Lerdahl and Jackendoff's [1983] Generative Theory of Tonal Music (GTTM). GTTM time span reduction (TSR) examines each note's musicological place in all hierarchical levels. The tempo curves are then generated by a parameterized parabolic equation which takes as input the result of the GTTM TSR analysis.

The performance has been 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 [1993] claimed that through informal listening tests in general the performances do not *sound* convincing.

The constraint of utilizing 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.

3.1.3. Composer Pulse and Predictive Amplitude Shaping. Manfred Clynes' Composer Pulse [Clynes 1986] also acts on multiple levels of the hierarchy. Clynes hypothesized 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 dynamic values in the score. It is hierarchical with separate values for within the beat, the phrase, and at multiple bar levels. Table IV shows the values of pulses for phrase

Table IV. Level 2 Composers' Pulses Multipliers—4-Pulse

Beethoven	Duration Amplitude	106 1.00	89 0.39	96 0.83	111 0.81
Mozart	Duration Amplitude	105 1.00	95 0.21	105 0.53	95 0.23
Schubert	Duration Amplitude	97 1.00	114 0.65	98 0.40	90 0.75
Haydn	Duration Amplitude	108 1.00	94 0.42	97 0.68	102 1.02
Schumann	Duration Amplitude	96 0.60	116 0.95	86 0.50	102 1.17
Mendelssohn	Duration Amplitude	118 1.00	81 0.63	95 0.79	104 1.12

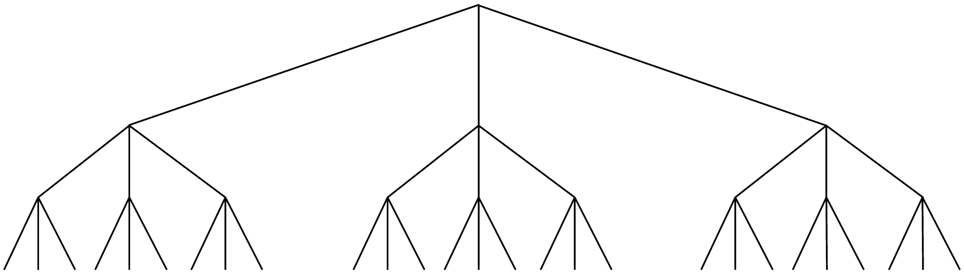

Fig. 3. Structure of a pulse set in three-time.

level for some composers. The pulses were measured using a sentograph to generate pressure curves from musicians tapping their finger while thinking of or listening to a specific composer. Figure 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 dynamic 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."[1] A fixed envelope shape model is used (some constants are manually defined by the user), the main inputs being distance 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 [Clynes 1995] showed that a number of expert and non-expert listeners preferred music with a composer's pulse rather than with a different pulse. However, not all tests on Clynes' approach have supported a universal pulse for each composer [Thompson 1989; Repp 1990], 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's open section: Beethoven's *Eroica* symphony, Opus 55, Movement 4, and Brahms' *Violin Concerto*, Opus 77, Movement 1. The Beethoven piece scored low, but the Brahms piece came first in the open section (beating two pieces submitted by Pop-E— see Section 3.1.7). The generation of this piece could have involved

[1] http://www.microsoundmusic.com/features.html.

significant amounts of manual work. Also, because it was the open section, the pieces submitted by Pop-E were not the same as those submitted by SuperConductor—hence, like was not compared to like. SuperConductor also won the open section in RenCon 2004 with J. S. Bach's, *Brandenburg Concerto*, Number 5, D major, third movement. The only competitor included from this review was Rubato (Section 3.1.6) performing a Bach piece. It should be reemphasized 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 was designed to allow a user to control the expressive shaping of a MIDI performance, giving significant amounts of control. However, outside the context of SuperConductor, the pulse has little scope for performance creativity—though amplitude shaping does. The pulse and amplitude shaping do not explicitly address nonmonophonic music, though SuperConductor can be used to generate polyphonic performances.

3.1.4. Bach Fugue System. In the Bach Fugue System [Johnson 1991], 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 to 90% of the time, though the article did not report how many tests were run. The system works in the context of polyphony.

3.1.5. Trumpet Synthesis. The testing of three out of the last four CSEMPs reviewed has focused on the keyboard. This pattern will continue throughout this article—most CSEMPs focus on the piano because it is easier to collect and analyze data for the piano than for other instruments. One of the first nonpiano systems was Dannenberg and Derenyi's Trumpet Synthesis [Dannenberg and Derenyi 1998; Dannenberg et al. 1998]. The authors' primary interest was to generate realistic trumpet synthesis, and adding performance factors improves this synthesis. The system was not designed for performance creativity but for simulation. This trumpet system synthesizes 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 developed using a semimanual 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 derived and stored during the analysis-by-synthesis.

No test results have been 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 nonaccompanied synthesis sounds quite impressive, only being let down by a synthetic feel toward the end of the phrase—though the note-to-note expression (as opposed to the synthesis) consistently avoids sounding machinelike. In both accompanied examples, it becomes clear as the performances goes on that a machine is playing, particularly in faster passages. But once again note-to-note expression does not sound too mechanical. Despite the reasonably positive nature of these examples, there was no attempt to objectively qualify how good the Trumpet Synthesis system is.

3.1.6. Rubato. G. Mazzola, a mathematician and recognized jazz pianist, developed a mathematical theory of music [Mazzola 1994; 2002]. Music is represented in an abstract geometrical space whose coordinates include onset time, pitch, and duration. A score exists in this space, and expressive performances are generated by performing transformations on the space. The basis of these transformations are a series of "operators" which can be viewed as a very generalized 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 Hierachical Parabola Model (Section 3.1.2) or DISTALL (Section 3.4.6). 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 discussed 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* modeled using Rubato was submitted, and came in fourth in the open section (SuperConductor came in 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* Opus 15, Number 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.

3.1.7. Pop-E. Pop-E [Hashida et al. 2006], a polyphrase ensemble system, was developed by some of the team involved in MIS (Section 3.2.1). It applies expression features separately to each voice in a MIDI file, through a synchronization algorithm. The music analysis uses GTTM local-level rules, and utilizes beams and slurs in the score to generate note groupings. So the expressive representation is up to the 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 [Hashida et al. 2007], a Java implementation of the system, provides such tools for shaping new performances.

To deal with polyphony, synchronization points are defined at the note grouping start and end points in the *attentive part*. The attentive part is that voice which is most perceptually prominent to a listener. The positions of notes in all other nonattentive parts are linearly interpolated relative to the synchronization 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 synchronization system is similar to the methods in the DISTALL system for dealing with polyphony—see Section 3.4.6.) Pop-E won the RenCon 2005 compulsory section, beating Director Musices (Section 3.1.1). In RenCon 2006, Pop-E won the compulsory section beating Kagurame (Section 3.4.2) and Ha-Hi-Hun (Section 3.4.3). In the open section in RenCon 2006, SuperConductor (Section 3.1.3) beat Pop-E with one performance, and lost to Pop-E with another.

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 [Sethares 2004]. Most systems in this review 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: for example, 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,” as noted an the first page of a 2003 marketing article on the Hermode Web site.[2] 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 optimized 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 Web site,[2] and the system has been successful enough to be embedded in a number of commercial products—for example, in Apple Logic Pro 7.

3.1.9. Sibelius. As mentioned in the introduction of this article, 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 [Finn 2007]. For dynamics, beat groups such as barlines, subbar 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 dynamic expression is constrained not to 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. In a personal communication, Finn reported 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’s 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.

[2] `http://www.hermode.com.`

3.1.10. Computational Music Emotion Rule System. In relation to the philosophy behind the Computational Music Emotion Rule System (CMERS), Livingstone et al. [2007], observed 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" (p. 47). 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 humanlike 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 two-dimensional model of human emotion space with four quadrants going from very active and negative to very active and positive, to very passive and positive to 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 included moving between major and minor modes, changing note pitch classes, and rising DM-type rules for small changes in dynamics and tempo. It was found that the addition of the microfeature humanization rules improved the accuracy of the emotional expression (as opposed to solely using macrofeature "recomposition" rules). The rules for humanizing the performance included some rules which are similar to those for Director Musices, such as phrase arch and emphasizing metrically important beats. Creative performance is possible in CMERS by adjusting the parameters of the rule set, and the emotional specification allows 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 review —one reason being that its aim is more measurable than a purely aesthetic goal. Examples of CMERS are available on this author's Web page.

3.2. Linear Regression

Learning CSEMPs can incorporate more knowledge more quickly than nonlearning 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 nonexpressive 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, for expressive performance of music, a linear model is almost certainly an oversimplification.

3.2.1. Music Interpretation System. The Music Interpretation Systems (MIS) [Katayose et al. 1990; Aono et al. 1997; Ishikawa et al. 2000] generates expressive performances in MIDI format, but learns 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 on to performance deviation actions. Its expressive representation is on the note and phrase level. MIS has methods to include some nonlinearities using logical ANDs between music features in the score, and a way of reducing redundant music features from its equations. This redundancy reduction improves its “generalization” ability (the ability for the system to perform music or composers that weren’t 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 [Meyer 1957] and Narmour’s [1990] Implication-Realization (IR) Model. 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 for a certain 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 synthesize the second half. Correlations (accuracies when compared to a human performance of the second half) were as follows: for velocity 0.87, for tempo 0.75, and for duration 0.92. A polyphonic MIS interpretation of Chopin, Opus 64, Number 2, was submitted to RenCon 2002. It came in third behind DISTALL (Section 3.4.6), beating three of the other four automated systems—DM, Kagurame (3.4.2), and Ha-Hi-Hun (3.4.3).

3.2.2. CaRo. CaRo [Canazza et al. 2000a, 2000b, 2001b, 2003, 2004] is a monophonic CSEMP designed to generate audio files which—like CMERS (Section 3.1.1)—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 interonset interval, brightness, and loudness-envelope centroid. A linear model is used to learn actions—every action has an equation characterized 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 generalization potential.

CaRo also learns “how musical performances are organised in the listener’s mind” [Canazza et al. 2001b, p. 35] in terms of moods: hard, heavy, dark, bright, light, and soft. To do this, a set of listening experiments analyzed by principal-components analysis (PCA) generates 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 the 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 modeling of expressive mood performances as realized by human performers.

3.3. Artificial Neural Networks

3.3.1. Artificial Neural Network Piano System. The earliest artificial neural network (ANN) approach was the Artificial Neural Network Piano System [Bresin 1998]. It has two incarnations. The first did not learn from human performers: a set of seven monophonic Director Musice 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 nonexpressive 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 generalized 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.

3.3.2. Emotional Flute. The Camurri et al. [2000] Emotional Flute system uses explicit music features and artificial neural networks, thus allowing greater generalization than the related CaRo system (Section 3.2.2). The music features are similar to those used in Director Musices. This CSEMP is strongly related to Bresin's second ANN in the previous section, extending it into the nonpiano realm and adding mood space modeling. Expressive actions include interonset 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* number 2 in nine different moods: cold, natural, gentle, bright, witty, serious, restless, passionate, and dark. Like CaRo (Section 3.2.2), a 2D mood space is generated and mapped on to the performances by the ANNs, and this mood space can be utilized 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.

3.4. Case and Instance-Based Systems

3.4.1. SaxEx. Arcos and Lopez de Mantaras' SaxEx [Arcos et al. 1997, 1998; Arcos and deMantaras 2001; Arcos and Lopez deMantaras 2001; Lopez deMantaras 2002] was one of the first systems to learn performances based on the performance context of mood. Like the trumpet system described earlier (Section 3.1.5), 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 nonexpressive audio file to perform transformations upon. Narmour's [1990] IR theory is used to analyze the music. Other elements used to analyze 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 nonexpressive 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 nonexpressive audio file and told to play it with a certain mood. A further artificial intelligence (AI) method is then used 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 case-based reasoning (CBR) solutions can be done automatically or manually through a graphical user interface (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 as in the Computational Music Emotion Rule System (Section 3.1.10)—the potential for the user to generate a performance with certain moods at different points in the music.

There has been no formal testing reported, but SaxEx examples are available online. The authors reported have "dozens" of positive comments about the realism of the music from informal listening tests, but no formal testing has been 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.

3.4.2. Kagurame. Kagurame [Suzuki et al. 1999; Suzuki 2003] 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 designed as a simulation system rather than for creative performance.

Results have been 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" [Suzuki et al. 1999, p. 647]. A high percentage of listeners guessed correctly whether the computer piece was in Romantic or Classical style. In RenCon 2004, Kagurame came in fourth in one-half of the compulsory section, one ahead of Director Musices, but was beaten by DM in the second half, coming in fifth. At RenCon 2006, a polyphonic performance of Chopin's piano etude in E major came in second—with Pop-E (Section 3.1.7) taking first place.

3.4.3. Ha-Hi-Hun. Ha-Hi-Hun [Hirata and Hiraga 2002] utilizes data structures designed to allow natural language statements to shape performance conditions (these include data structures to deal with nonmonophonic music). Hirata and Hiraga's [2002] article focused 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 piece Y bearing in mind that it will be used to generate creative performances of the X piece. 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 Y piece 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 outward. 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 have been given. In RenCon 2002, Ha-Hi-Hun learned to play the Chopin etude, Opus 10, Number 3, through learning the style of a human performance of Chopin's nocturne, Opus 32, Number 2. The performance came ninth out of 10 submitted performances by other CSEMPs (many of which were manually produced). In RenCon 2004, Ha-Hi-Hun came in last in the compulsory section, beaten by both Director Musices and Kagurame (Section 3.4.2). In RenCon 2006, a performance by Ha-Hi-Hun also came in third out of six in the compulsory section, beaten by Pop-E (Section 3.1.7) and Kagurame (Section 3.4.2).

3.4.4. PLCG System. Gerhard Widmer has applied various versions of a rule-based learning approach, attempting to utilize a larger database of music than previous CSEMPs. The PLCG system [Widmer 2000, 2002, 2003] 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 *metalearning* algorithm—the underlying algorithm being Sequential Covering [Mitchell 1997]. 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 PLCG, including its generalization ability. Widmer's systems are the only CSEMPs reviewed here that have had any significant generalization testing. The testing methods were based on correlation approaches. Seven pieces selected from the scores used in learning were regenerated using the rule set, and their tempo/dynamics profiles compared to the original performances very favorably. Regenerations were compared to performances by a different human performer, Phillipe 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.

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 [Widmer and Tobudic 2003]. 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 provided 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—Section 3.1.2).

Once the curve-fitting has been done, there is still a "residual" expression in the human MIDI performance. This is hypothesized 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 our review.

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 nonexpressive 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.

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 [Tobudic and Widmer 2003a; 2003b; 2003c; 2005] 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 [Emde and Wettschereck 1996]). 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 experiments similar to those described in Section 3.4.5. 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 was set equal to 5% of the main note's duration, and the 5% of duration was divided equally among the grace notes. The performance was the top-scored automated performance at RenCon 2002—ahead of Kagurame (Section 3.4.2), MIS (Section 3.2.1), and Ha-Hi-Hun (Section 3.4.3)—and it beat one nonautomated system.

3.5. Statistical Graphical Models

3.5.1. Music Plus One. The Music Plus One system [Raphael 2001a, 2001b, 2003] 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 Web site 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 (3.4.3), Kagurame (3.4.2), and Widmer's system (3.4.6). 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 behavior 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 seven notes while 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.

3.5.2. ESP Piano System. Grindlay's [2005] ESP Piano system is a polyphonic CSEMP designed to simulate expressive playing of pieces of piano music which consist of largely monophonic melodies, 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's 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 were discussed by the author. The accompaniment is analyzed 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 chord (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 is 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, discussed in Section 3.4.2). 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.

3.6. Other Regression Methods

3.6.1. Drumming System. Thus far in this review only pitched instrument systems have been surveyed—mainly piano, saxophone, and trumpet. A system for nonpitched (drumming) expression will now be examined. In the introduction it was mentioned that pop music has enthusiastically utilized the "robotic" aspects of MIDI sequencers. However, eventually some pop musicians wanted a more realistic sound for their electronic music, and *humanization* 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; see Section 3.1.3). Such groove systems have been applied commercially in mass-market systems like Propellorhead Reason, where it is possible to generate *groove templates* from a drum track and apply them to any other MIDI track [Carlson et al. 2003]. However, just as some research has suggested limitations in the application of Clynes' composer pulses, so Wright and Berdahl's [2006] research showed the limits of groove templates. Their analysis of multivoiced Brazilian drumming recordings found that groove templates could only account for 30% of expressive timing.

Wright and Berdahl [2006] investigated other methods to capture the expressive timing using a system that learns from audio files. The audio features examined were a note's timbre, metric position, and rhythmic context, the rhythmic context being the timbres and relative temporal position of notes within one beat of the input notes—thus giving the system a narrow expressive representation. The system learns to map these audio features onto the timing deviations of each nonpitched note; it is not designed to generate creative performances, but to simulate them. The mapping model is based on regression between audio features and timing deviation (vs. a quantized version of the beat). Three different methods of learning the mapping model were tried: Kernel Ridge Regression [Hastie et al. 2001], Gaussian Process Regression [Rasmussen and Williams 2006], and kNN [Hellman 1991]. This learning approach was found to track the expressive timing of the drums much better than the groove templates, clearly demonstrated in their graphs showing the error over the drum patterns. All three learning methods were found to give approximately equal accuracy, though it is believed that Gaussian Process Regression has the greatest room for improvement. Examples are provided online. Note that the system is not limited only to Brazilian drumming; Wright and Berdahl [2006] also tested it on Reggae rhythms with similar success.

3.6.2. KCCA Piano System. The most recent application of kernel regression methods to expressive performance is the system by Dorard et al. [2007]. Their main aim was 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 a "worm" representation of expressive performance [Dixon et al. 2002]. The worm is a visualization tool for the loudness and tempo aspects of expressive performance. It uses a 2D 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 color fades and size decrease 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 [2007] 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) [Sholkopf et al. 1998] 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 (see the artificial neural network models discussed in Section 3.3, Sibelius discussed in Section 3.1.9, and also the discussion in Section 3.7.3).

Testing was performed on Chopin's *Etude 3*, Opus 10—the system was trained on the worm of the first eight bars, and then tried to complete the worm for bars 9 to 12. The correlation between the original human performance worm for 9 to 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 not sound very realistic.

3.7. Evolutionary Computation

A number of more recent CSEMPs have used evolutionary computation [EC] methods, such as genetic algorithms [Mitchell 1996] or Multiagent systems [Kirke 1997]. In general (but not always), such systems have opportunities for performance creativity. They often have a parameterization that is simple to change—for example, a fitness function. They also have an emergent [Chalmers 2006] output which can sometimes produce unexpected but coherent results.

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 [Ramirez and Hazan 2005]. 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 C4.5 [Quinlan 1993]. This was built for musicological purposes—to see what kinds of rules were generated—not to generate any performances. The decision tree system had a three-note-level expressive representation, and music features used to characterize a note included metrical position and some Narmour [1990] 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 tree rather than a 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 [Witten and Frank 2000] 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 modeling methods were tried (linear regression and four different forms of support vector machines) but they didn't fare as well correlationwise.

Ramirez and Hazan's next system [Hazan and Ramirez 2006] 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 article [Ramirez and Hazan 2005], 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 were enhanced because, in addition to modeling timing and dynamics, the trees modeled 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 relation to jazz saxophone. It was possible to examine these deviations because the fitness function was implemented using an edit distance [Levenshtein 1966] 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 Ramirez and Hazan [2005] said, "performance is an inexact phenomenon" (p. 31). 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.

3.7.2. Sequential Covering Algorithm GAs. The Sequential Covering Algorithm Genetic Algorithm (GA) [Ramirez and Hazan 2007] 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. Ramirez and Hazan [2007] returned to their first (non-EC) article's level of expressive representation—looking at note-level deviations *without* ornamentation or consolidation. However, they made significant improvements over their original non-EC article. The correlation coefficients for onset, duration, and energy/loudness in the original system were 0.72, 0.44, and 0.67, respectively, but in this new system they were 0.75, 0.84, and 0.86, respectively—significantly higher. And this system also has the advantage of slightly greater creativity due to its GA approach.

3.7.3. Generative Performance GAs. A parallel thread of EC research is Zhang and Miranda's [2006a; 2006b] monophonic generative performance GAs which evolve pulse sets (see Section 3.1.3). Rather than comparing the generated performances to actual performances, the fitness function here expresses constraints inspired by the generative performance work of Eric Clarke [1998]. 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, increases the performance creativity of the system. The timing curve comes from an algorithm based on Cambouropoulos' [2001] 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 [1991]), the metrical strength of a note (based on the Melisma software [Temperley and Sleator 1999]), 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; (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 dynamic 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. Point (3) is reminiscent of the restriction of expression used in the ANN models (Section 3.3.1) and the KCCA Piano (Section 3.6.2) model described earlier. It is designed to prevent the deviations from becoming too extreme.

There has been no formal testing of this GA work, though the authors demonstrated—using as an example 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 deviations curves. Zhang and Miranda [2006a] claimed that "when listening to pieces performed with the evolved pulse sets, we can perceive the expressive dynamics of the piece" (p. 1839). However, more evaluation would be helpful; repeating pulse sets have been shown not to be universally applicable as an expressive action format. Post-Clynes CSEMPs have shown more success using *noncyclic* expression.

3.7.4. Multi-Agent System with Imitation. Zhang and Miranda [2007] developed the above into a MultiAgent System (MAS) with Imitation—influenced by Miranda's [2002] evolution of music MAS study, and inspired by the hypothesis that expressive music performance strategies emerge through interaction and evolution in the performers' society. 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 modeled as a hierarchical pulse set. So it has the same expressive representation. When an agent hears a performance, it evaluates it as being better than its own, it attempts to imitate it but ends up giving a slightly mutated performance. This imitating agent then generates a pulse set internally that matches its mutated performance. So pulse sets are not exchanged, performances are (just as humans imitate behavior, not neural and underlying biological processes). The performances of the system, which were generated through 100 generations of imitation, were once again not evaluated by the researchers, though it was found that the imitation approach generated performances more slowly than the previous GA approach.

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 can 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.

This Zhang/Miranda MAS system has much potential for future work. For example, an approach could be investigated that allowed each agent to have its own evaluation function. Some agents could use comparisons with human performances as an evaluation function, while some agents could be equipped with entirely novel non-human-fitness evaluation functions. Thus, by altering the proportion of agents with different types of evaluation functions, different types of performances could be generated in the population. Another possibility would be to allow more direct user intervention in shaping performances by letting users become agents in the system themselves.

3.7.5. Ossia. Like the Computational Music Emotion Rule System (Section 3.1.10), Dahlstedt's [2007] 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 nonlearning systems (the same reason applies for the system described in Section 3.7.6). 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 in which 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 Dannenberg [1993] and Anders [2007]. 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 review 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 the musical structure. More flexible representations than MIDI include MusicXML, ENP-score-notation [Laurson and Kuuskankare 2003], WEDELMUSIC XML [Bellini and Nesi 2001], MusicXML4R [Good 2006], and the proprietary representations used by commercial software such as Sibelius, Finale, Notion, and Zenph High-Resolution MIDI [Atkinson 2007] (which was recently used on a CD release of automated Disklavier reperformances of Glenn Gould).

Many of the performance systems described in this review 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 emphasizing 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, repetivity, amount of silence, pitch variation, and level of recursion. These fitness rules were developed heuristically by Dahlstedt [2007] through analysis-by-synthesis methods.

Ossia's level of expressive representation is equal to its level of compositional representation. Dahlstedt [2007] observed: "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 Web site,[3] including a composed suite. Dahlstedt [2007] claimed 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" (p. 9). The examples on the Web site 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., by a preference for pre-1940s classical music, or pop music). Another approach is musicological analysis, but the problem then becomes that musicological

[3] http://www.init.ituniv.se/~palle/pmwiki/pmwiki.php?n=Works.Ossia.

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 on the Web site 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 have 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 Barret [Toop 1988]. 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 beings 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 review.) 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 (Section 3.1.10), the composition/performance dichotomy starts to break down—the dichotomy is really between macrofeatures and microfeatures of the music.

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 goes toward analyzing the musical structure of the score/audio. 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—that is, 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 to 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 utilize 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 multispeaker 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 machinelike. In Zhang/Miranda's system (Section 3.7.4) the agents imitate each others' 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 place of A performing its tune.

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 *nonexcited* agent, making macroerrors 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 are part of the "original" composition. So the listening agent attempts to memorize the tune as it hears it, including any performance errors or changes. As the agents perform to each other, they store internally 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. 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 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; 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 "superphrase" 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 supersuper-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 is 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.

4. SUMMARY

Before reading this summary, another viewing of Table I at the start of the review may be helpful to the reader. Expressive performance is a complex behavior with many causative conditions, so it is no surprise that in this review 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 noncustom method for analysis of music features was GTTM, followed by IR. Due to its simplicity in modeling performance, the most common instrument simulated was piano—but interestingly this was followed closely by saxophone, possibly because of the popularity

Table V. Summary of Four Primary Terms of Reference

CSEMP	Expressive Representation	Testing Status	Nonmonophonic	Performance Creativity
Director Musices	7	10	Yes	8
Hierarchical Parabola Model	8	7		3
Composer Pulse	6	9	Yes	5
Bach Fugue	6	4	Yes	3
Rubato	7	4	Yes	8
Trumpet Synthesis	4	1		3
MIS	6	6	Yes	3
ANN Piano	4	6		3
Music Plus One	4	7	Yes	4
SaxEx	5	1		8
CaRo	3	6		7
Emotional Flute	4	6		6
Kagurame	8	8	Yes	3
Ha-Hi-Hun	6	6	Yes	8
PLCG	4	6		3
Phrase-Decomposition/PLCG	10	6		3
DISTALL	10	9	Yes	3
Pop-E	6	10	Yes	8
ESP Piano	6	6	Yes	3
KCCA Piano	4	6	Yes	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	Yes	10
CMERS	6	6	Yes	10
pMIMACS	8	1		10

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 review, 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. Also, about half of the systems have only been tested on monophonic tunes. Polyphony and homophony introduce problems both in terms of synchronization and music feature analysis. Further, in regard 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 summarized in Table V. 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 submission, formal listening,

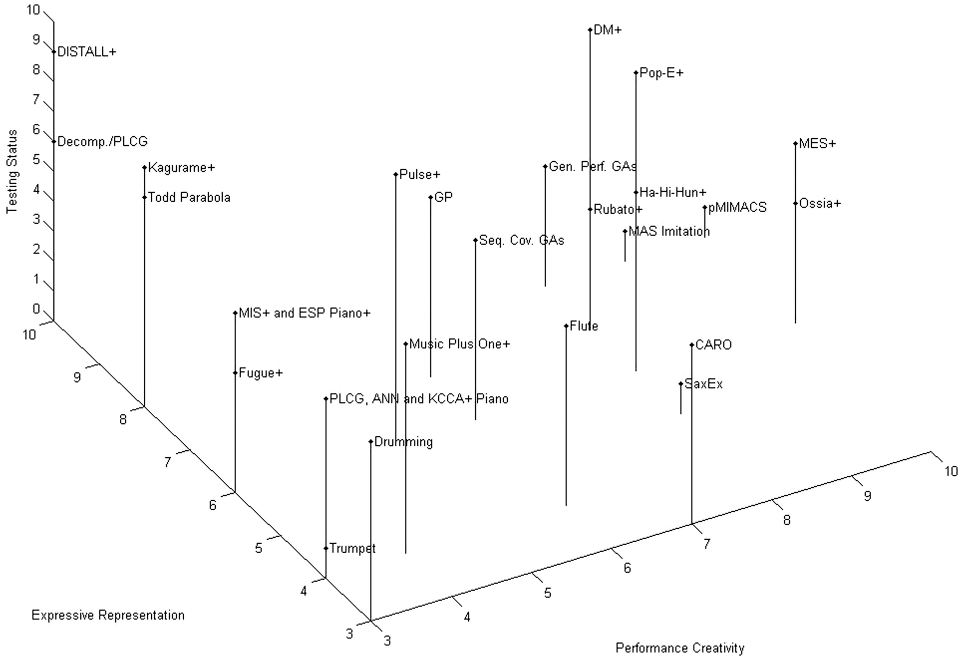

Fig. 4. Evaluation of reviewed systems relative to terms of reference ("+" means polyphonic).

correlation, and/or successful utilization 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 useful indicators for anyone wanting an overview of the field, based on the four elements discussed at the start of this article. Figure 4 shows a 3D plot of summary Table V.

5. CONCLUSIONS

There have been significant achievements in the field of simulating human musical performance in the last 25 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 forward. 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 [Hamanaka et al. 2005]. 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 for efficiency, controllability, and creativity. This review began by developing four terms of reference 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-fertilize each other: performance algorithms for expressing structure and emotion/mood can help composition, as well as composition providing more creative and controlled computer performance. The question is also open as to what forms of nonhuman 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 modeling of performance should go beyond ANNs and instrument physical modeling. The human/instrument performance process is a complex dynamical system for which there have been some deeper psychological and physical studies. 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 [Koelsch and Siebel 2005; Britton et al. 2006; Durrant et al. 2007], and Eric Clarke [1993] has written on the importance of physical embeddedness of human performance. But, although researchers such as Parncutt [1997] (in his virtual pianist approach) and Widmer [Widmer and Goebl 2004] have highlighted the opportunity for deeper models, there has been little published progress in this area of the CSEMP field.

So to conclude: the overarching focus so far means that there are opportunities for some better-tested, more creative, neurological and biomechanical models of human performance. These will be systems which not only help to win the Chopin contest, but to utilize 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 microspecification domain, and more easily and accurately generate expressive performances of their work. And the music industry will be able to expand the power of humanization tools, creating new efficiencies in recording and performance.

REFERENCES

ANDERS, T. 2007. Composing music by composing rules: Design and usage of a generic music constraint system. Ph.D. dissertation. University of Belfast, Belfast, Northern Ireland.

AONO, Y., KATAYOSE, H., AND INOKUCHI, S. 1997. Extraction of expression parameters with multiple regression analysis. *J. Inform. Process. Soc. Japan 38*, 1473–1481.

ARCOS, J. L., DE MANTARAS, R. L., AND SERRA, X. 1997. SaxEx: A Case-based reasoning system for generating expressive musical performances. In *Proceedings of 1997 International Computer Music Conference*, Thessalonikia, Greece, September 1997, P. R. Cook, Ed. ICMA, San Francisco, CA, 329–336.

ARCOS, J. L., LOPEZ DE MANTARAS, R., AND SERRA, X. 1998. Saxex: A case-based reasoning system for generating expressive musical performance. *J. New Music Res. 27*, 194–210.

ARCOS, J. L. AND DE MANTARAS, R. L. 2001. The SaxEx system for expressive music synthesis: A progress report. In *Proceedings of the Workshop on Current Research Directions in Computer Music* (Barcelona, Spain, November), C. Lomeli and R. Loureiro, Eds. 17–22.

ARCOS, J. L. AND LOPEZ DE MANTARAS, R. 2001. An interactive case-based reasoning approach for generating expressive music. *J. Appl. Intell. 14*, 115–129.

ATKINSON, J. 2007. J. S. Bach: *The Goldberg Variations*. *Stereophile*, September 2007.

BELLINI, P. AND NESI, P. 2001. WEDELMUSIC format: An XML music notation format for emerging applications. In *Proceedings of the 1st International Conference on Web Delivering of Music*. Florence, Italy, November. IEEE Press, Los Alamitos, CA, 79–86.

BRESIN, R. 1998. Artificial neural networks based models for automatic performance of musical scores. *J. New Music Res. 27*, 239–270.

BRESIN, R. AND FRIBERG, A. 2000. Emotional coloring of computer-controlled music performances. *Comput. Music J. 24*, 44–63.

BRITTON, J. C., PHAN, K. L., TAYLOR, S. F., WELSCH, R. C., BERRIDGE, K. C., AND LIBERZON, I. 2006. Neural correlates of social and nonsocial emotions: An fMRI study. *NeuroImage 31*, 397–409.

BUXTON, W. A. S. 1977. A composer's introduction to computer music. *Interface 6*, 57–72.

CAMBOUROPOULOS, E. 2001. The Local Boundary Detection Model (LBDM) and its application in the study of expressive timing. In *Proceedings of the 2001 International Computer Music Conference*, Havana, Cuba, September 2001, R. Schloss and R. Dannenberg, Eds. International Computer Music Association, San Fransisco, CA.

CAMURRI, A., DILLON, R., AND SARON, A. 2000. An experiment on analysis and synthesis of musical expressivity. In *Proceedings of the 13th Colloquium on Musical Informatics* (L'Aquila, Italy, September).

CANAZZA, S., DE POLI, G., DRIOLI, C., RODA, A., AND ZAMPERINI, F. 2000a. Real-time morphing among different expressive intentions on audio playback. In *Proceedings of the International Computer Music Conference* (San Francisco, CA, Aug.). 356–359.

CANAZZA, S., DRIOLI, C., DE POLI, G., RODÀ, A., AND VIDOLIN, A. 2000b. Audio morphing different expressive intentions for multimedia systems. *IEEE Multimed. 7*, 79–83.

CANAZZA, S., DE POLI, G., DRIOLI, C., RODÀ, A., AND VIDOLIN, A. 2001a. Expressive morphing for interactive performance of musical scores. In *Proceedings of 1st International Conference on WEB Delivering of Music* (Florence, Italy, Nov.). IEEE Press, Los Alamitos, CA, 116–122.

CANAZZA, S., DE POLI, G., RODÀ, A., VIDOLIN, A., AND ZANON, P. 2001b. Kinematics-energy space for expressive interaction on music performance. In *Proceedings of the MOSART Workshop on Current Research Directions on Computer Music* (Barcelona, Spain, Nov.). 35–40.

CANAZZA, S., DE POLI, G., RODÀ, A., AND VIDOLIN, A. 2003. An abstract control space for communication of sensory expressive intentions in music performance. *J. New Music Res. 32*, 281–294.

CANAZZA, S., DE POLI, G., DRIOLI, C., RODÀ, A., AND VIDOLIN, A. 2004. Modeling and control of expressiveness in music performance. *Proc. IEEE 92*, 686–701.

CARLSON, L., NORDMARK, A., AND WIKLANDER, R. 2003. *Reason Version 2.5—Getting Started*. Propellorhead Software, Stockholm, Sweden.

CHALMERS, D. 2006. Strong and weak emergence. In *The Re-Emergence of Emergence*, P. Clayton and P. Davies, Eds. Oxford University Press, Oxford, U.K.

CHURCH, M. 2004. The mystery of Glenn Gould. *The Independent*, Nov. 29.

CLARKE, E. F. 1993. Generativity, mimesis and the human body in music performance. *Contemp. Music Rev. 9*, 207–219.

CLARKE, E. F. 1998. Generative principles in music performance. In *Generative Processes in Music: The Psychology of Performance, Improvisation, and Composition*, J. A. Sloboda, Ed. Clarendon Press, Oxford, U.K., 1–26.

CLARKE, E. F. AND WINDSOR, W. L. 2000. Real and simulated expression: A listening study. *Music Percept. 17*, 277–313.

CLYNES, M. 1986. Generative principles of musical thought: Integration of microstructure with structure. *Commun. Cognit. 3*, 185–223.

CLYNES, M. 1995. Microstructural musical linguistics: Composer's pulses are liked best by the musicians. *Cognition: Int. J. Cogn. Sci. 55*, 269–310.

COPE, D. 1992. Computer modelling of musical intelligence in experiments in musical intelligence. *Comput. Mus. J. 16*, 69–83.

DAHLSTEDT, P. 2007. Autonomous evolution of complete piano pieces and performances. In *Proceedings of the ECAL Workshop on Music and Artificial Life* (MusicAL , Lisbon, Portugal, Sep.), 10.

DANNENBERG, R. B. 1993. A brief survey of music representation issues, *Tech. Syst. Comput. Music J. 17*, 20–30.

DANNENBERG, R. B. AND DERENYI, I. 1998. Combining instrument and performance models for high-quality music synthesis. *J. New Music Res. 27*, 211–238.

DANNENBERG, R. B., PELLERIN, H., AND DERENYI, I. 1998. A study of trumpet envelopes. In *Proceedings of the International Computer Music Conference*, Ann Arbor, Michigan, October 1998. International Computer Music Association, San Francisco, 57–61.

DESAIN, P. AND HONING, H. 1993. Tempo curves considered harmful. *Contemp. Music Rev. 7*, 123–138.

DIXON, S., GOEBL, W., AND 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, Sep.). 361–364.

Dorard, L., Hardoon, D. R., Shawe-Taylor, J. 2007. Can style be learned? A machine learning approach towards 'performing' as famous pianists. In *Proceedings of the Music, Brain and Cognition Workshop—NIPS* (Whistler, Canada).

Durrant, S., Miranda, E. R., Hardoon, D., Shawe-Taylor, J., Brechmann, A., and Scheich, H. 2007. Neural correlates of tonality in music. In *Proceedings of Music, Brain & Cognition Workshop—NIPS Conference*.

Emde, W. and Wettschereck, D. 1996. Relational instance based learning. In *Proceedings of 13th International Conference on Machine Learning*, Bari, Italy, July 1996, L. Saitta, Ed. Morgan Kaufmann, San Francisco, CA, 122–130.

Finn, B. 2007. Personal communication. August.

Friberg, A. 1995. A quantitative rule system for music performance. Ph.D. dissertation. Department of Speech, Music and Hearing, Royal Institute of Technology, Stockholm, Sweden.

Friberg, A. 2006. pDM: An expressive sequencer with real-time control of the KTH music-performance rules. *Comput. Music J. 30*, 37–48.

Friberg, A., Bresin, R., and Sundberg, J. 2006. Overview of the KTH rule system for musical performance. *Adv. Cognit. Psych. 2*, 145–161.

Friberg, A. and J. Sundberg. 1999. Does music performance allude to locomotion? A model of final ritardandi derived from measurements of stopping runners. *J. Acoust. Soc. Amer. 105*, 1469–1484.

Gabrielsson, A. 2003. Music performance research at the millenium. *Psychol. Music 31*, 221–272.

Gabrielsson, A. and Juslin, P. 1996. Emotional expression in music performance: Between the performer's intention and the listener's experience. *Psychol. Music 24*, 68–91.

Good, M. 2001. MusicXML for notation and analysis. In *The Virtual Score: Representation, Retrieval, Restoration*, W. B. Hewlett and E. Selfridge-Field, Eds. MIT Press, Cambridge, MA, 113–124.

Good, M. 2006. MusicXML in commercial applications. In *Music Analysis East and West*, W. B. Hewlett and E. Selfridge-Field, Eds. MIT Press, Cambridge, MA, 9–20.

Grindlay, G. C. 2005. Modeling expressive musical performance with Hidden Markov Models. Ph.D. dissertation. University of California, Santa Cruz, Santa Cruz, CA.

Hamanaka, M., Hirata, K., and Tojo, S. 2005. ATTA: Automatic Time-span Tree Analyzer based on extended GTTM. In *Proceedings of the 6th International Conference on Music Retrieval Conference* (London, U.K., Sep.). 358–365.

Hashida, M., Nagata, N., and Katayose, H. 2006. Pop-E: A performance rendering system for the ensemble music that considered group expression. In *Proceedings of the 9th International Conference on Music Perception and Cognition*, Bologna, Spain, August 2006, M. Baroni, R. Addessi, R. Caterina, and M. Costa, Eds. ICMPC, 526–534.

Hashida, M., Nagata, N., and Katayose, H. 2007. jPop-E: An assistant system for performance rendering of ensemble music. In *Proceedings of the Conference on New Interfaces for Musical Expression (NIME'07)*, L. Crawford, Ed. 313–316.

Hastie, T., Tibshirani, R., Friedman, J. 2001. *The Elements of Statistical Learning*. Springer, Berlin, Germany.

Hazan, A. and Ramirez, R. 2006. Modelling expressive performance using consistent evolutionary regression trees. In *Proceedings of 17th European European Conference on Aritifial Intelligence (Workshop on Evolutionary Computation)*, Riva del Garda, Italy, August 2006, G. Brewka, S. Coradeschi, A. Perini, and P. Traverso, Eds. IOS Press, Amsterdam, The Netherlands.

Hellman, M. 1991. The nearest neighbor classification rule with a reject option. In *Nearest Neighbor (NN) Norms: Pattern Classification Techniques*, B. Dasarathy, Ed. IEEE Computer Society Press, Los Alamitos, CA.

Hiller, L. and Isaacson, L. 1959. *Experimental Music. Composition with an Electronic Computer*. McGraw Hill, New York, NY.

Hiraga, R., Bresin, R., and Hirata, K. and K. H. 2004. Rencon 2004: Turing test for musical expression. In *Proceedings of the New Interfaces for Musical Expression Conference*, Hamatsu, Japan, June, Y. Nagashima and M. Lyons, Eds. Shizuoka University of Art and Culture, Hamatsu, Japan, 120–123.

Hirata, K. and 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), 40–46.

Ishikawa, O., Aono, Y., Katayose, H., and 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, CA, 348–351.

JOHNSON, M. L. 1991. Toward an expert system for expressive musical performance. *Comput. 24*, 30–34.

JUSLIN, P. 2003. Five facets of musical expression: A psychologist's perspective on music performance. *Psychol. Music* 31, 273–302.

KAJANOVA, Y. 2002. Johann Sebastian Bach and the modern jazz quartet—a desire for sonic sensualism or seriousness. In *Bach 2000. Music Between Virgin Forest and Knowledge Society.* J. Fukac, Ed. Compostela Group of Universities, 253–260.

KATAYOSE, H., FUKUOKA, T., TAKAMI, K., AND INOKUCHI, S. 1990. Expression extraction in virtuoso music performances. In *Proceedings of the 10th International Conference on Pattern Recognition* (Atlantic City, NJ, June). IEEE Press, Los Alamitos, CA, 780–784.

KIRKE, A. 1997. Learning and co-operation in mobile multi-robot systems. Ph.D. dissertation, University of Plymouth, Plymouth, U.K.

KIRKE, A. AND MIRANDA, E. R. 2007a. Capturing the aesthetic: Radial mappings for cellular automata music. *J. ITC Sangeet Res. Acad. 21*, 15–23.

KIRKE, A. AND MIRANDA, E. R. 2007b. Evaluating mappings for cellular automata music. In *Proceedings of ECAL Workshop on Music and Artificial Life* (MusicAL, Lisbon, Portugal).

KOELSCH, S. AND SIEBEL, W. A. 2005. Towards a neural basis of music perception. *Trends Cognit. Sci. 9*, 579–584.

KROISS, W. 2000. Parameteroptimierung für ein Modell des musikalischen Ausdrucks mittels Genetischer Algorithmen. Master's thesis. Department of Medical Cybernetics and Artificial Intelligence, University of Vienna, Vienna, Austria.

KRUMHANS, C. 1991. *Cognitive Foundations of Musical Pitch*. Oxford University Press, Oxford, U.K.

LAURSON, M. AND KUUSKANKARE, M. 2003. From RTM-notation to ENP-score-notation. In *Proceedings of Journées d'Informatique Musicale* (Montbéliard, France).

LERDAHL, F. AND JACKENDOFF, R. 1983. *A Generative Theory of Tonal Music*. MIT Press, Cambridge, MA.

LEVENSHTEIN, V. I. 1966. Binary codes capable of correcting deletions, insertions, and reversals. *Sov. Phys. Dokl. 10*, 707–710.

LIVINGSTONE, S. R., MUHLBERGER, R., BROWN, A. R., AND LOCH, A. 2007. Controlling musical emotionality: An affective computational architecture for influencing musical emotions. *Dig. Creativ. 18*, 1 (Mar.), 43–53.

LOPEZ DE MANTARAS, R. AND ARCOS, J. L. 2002. AI and music: From composition to expressive performances. *AI Mag. 23*, 43–57.

MAZZOLA, G. 2002. *The Topos of Music—Geometric Logic of Concepts, Theory, and Performance*. Birkhäuser, Basel, Switzerland/Boston, MA.

MAZZOLA, G. AND ZAHORKA, O. 1994. Tempo curves revisited: Hierarchies of performance fields. *Comput. Music J. 18*, 1, 40–52.

MEYER, L. B. 1957. Meaning in music and information theory. *J. Aesthet. Art Crit. 15*, 412–424.

MIRANDA, E. R. 2001. *Composing Music With Computers*. Focal Press, Oxford, U.K.

MIRANDA, E. R. 2002. Emergent sound repertoires in virtual societies. *Comput. Music J. 26*, 2, 77–90.

MITCHELL, M. 1996. *An Introduction to Genetic Algorithms*. MIT Press, Cambridge, MA.

MITCHELL, T. 1997. *Machine Learning*. McGraw-Hill, New York, NY.

NARMOUR, E. 1990. *The Analysis and Cognition of Basic Melodic Structures: The Implication-Realization Model*. University of Chicago Press, Chicago, IL.

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.).

PALMER, C. 1997. Music performance. *Ann. Rev. Psychol. 48*, 115–138.

PACE, I. 2009. Notation, time and the performer's relationship to the score in contemporary music. In *Collected Writings of the Orpheus Institute: Unfolding Time: Studies in Temporality in Twentieth-Century Music*, D. Crispin, Ed. Leuven University Press, Leuven, Belgium, 151–192.

PAPADOPOULOS, G. AND WIGGINS, G. A. 1999. AI methods for algorithmic composition: A survey, a critical view, and future prospects. In *Proceedings of the AISB Symposium on Musical Creativity*.

PARNCUTT, R. 1997. Modeling piano performance: Physics and cognition of a virtual pianist. In *Proceedings of the 1997 International Computer Music Conference*, Thessalonikia, Greece, September 1997, P. R. Cook, Ed. ICMA, San Fransisco, CA, 15–18.

QUINLAN, J.R. 1993. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, San Francisco, CA.

Ramirez, R. and Hazan, A. 2005. Modeling expressive performance in jazz. In *Proceedings of the 18th International Florida ArtificiaI Intelligence Research Society Conference (AI in Music and Art)*, Clearwater Beach, FL, May 2005, AAAI Press, Menlo Park, CA, 86–91.

Ramirez, R. and Hazan, A. 2007. Inducing a generative expressive performance model using a sequential-covering genetic algorithm. In *Proceedings of the Genetic and Evolutionary Computation Conference*. (London, UK, July), ACM Press, New York, NY.

Raphael, C. 2001a. Can the computer learn to play music expressively? In *Proceedings of Eighth International Workshop on Artificial Intelligence and Statistics, 2001*, T. Jaakkola and T. Richardson, Eds. Morgan Kaufmann, San Franciso, CA, 113–120.

Raphael, C. 2001b. A Bayesian network for real-time musical accompaniment. *Neur. Inform. Process. Syst. 14*, 1433–1439.

Raphael, C. 2003. Orchestra in a box: A system for real-time musical accompaniment. In *Proceedings of the 2003 International Joint Conference on Artificial Intelligence (Working Notes of RenCon Workshop)*, Acapulco, Mexico, August 2003, G. Gottlob and T. Walsh, Eds. Morgan Kaufmann, San Francisco, CA, 5–10.

Rasmussen, C. and Williams, C. 2006. *Gaussian Processes for Machine Learning.* MIT Press, Cambridge, MA.

Repp, B. H. 1990. Composer's pulses: Science or art. *Music Percept. 7*, 423–434.

Roads, C. 1996. *The Computer Music Tutorial.* MIT Press, Cambridge, MA.

Seashore, C. E. 1938. *Psychology of Music*. McGraw-Hill, New York, NY.

Sethares, W. 2004. *Tuning, Timbre, Spectrum, Scale*. Springer, London, U.K.

Sholkopf, B., Smola, A., and Muller, K. 1998. Nonlinear component analysis as a kernel eigenvalue problem. *Neur. Computat. 10*, 1299–1319.

Sundberg, J., Askenfelt, A., and Frydén, L. 1983. Musical performance. A synthesis-by-rule approach. *Comput. Music J. 7*, 37–43.

Sundberg, J., Friberg, A., and Bresin, R. 2003. Attempts to reproduce a pianist's expressive timing with Director Musices performance rules. *J. New Music Res. 32*, 317–325.

Suzuki, T. 2003. Kagurame Phase-II. In *Proceedings of the 2003 International Joint Conference on Artificial Intelligence (Working Notes of RenCon Workshop)*, Acapulco, Mexico, August 2003, G. Gottlob and T. Walsh, Eds. Morgan Kauffman, San Francisco, CA.

Suzuki, T., Tokunaga, T., and 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, August 1999, Morgan Kaufmann, San Francisco, CA, 642–648.

Temperley, D. and Sleator, D. 1999. Modeling meter and harmony: A preference rule approach. *Comput. Music J. 23*, 10–27.

Thompson, W. F. 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.

Tobudic, A. and Widmer, G. 2003a. Relational IBL in music with a new structural similarity measure. In *Proceedings of the 13th International Conference on Inductive Logic Programming*, Szeged, Hungary, September 2003, T. Horváth and A. Yamamoto, Eds. Springer Verlag, Berlin, Germany, 365–382.

Tobudic, A. and Widmer, G. 2003b. Learning to play Mozart: Recent improvements. In *Proceedings of the IJCAI'03 Workshop on Methods for Automatic Music Performance and their Applications in a Public Rendering Contest (RenCon)*, Acapulco, Mexico, August, 2003. K. Hirata Ed.

Tobudic, A. and Widmer, G. 2003c. Technical notes for musical contest category. In *Proceedings of the 2003 International Joint Conference on Artificial Intelligence (Working Notes of RenCon Workshop)*, Acapulco, Mexico, August 2003, G. Gottlob and T. Walsh, Eds. Morgan Kauffman, San Francisco, CA.

Tobudic, A. and Widmer, G. 2005. Learning to play like the great pianists. In *Proceedings of the 19th International Joint Conference on Artificial Intelligence*, Edinburgh, UK, August 2005, P. Kaelbling and A. Saffiotti, Eds. Professional Book Center, Denver, CO, 871–876.

Todd, N. P. 1985. A model of expressive timing in tonal music. *Music Percept. 3*, 33–58.

Todd, N. P. 1989. A computational model of Rubato. *Contemporary Music Rev. 3*, 69–88.

Todd, N. P. 1992. The dynamics of dynamics: A model of musical expression. *J. Acoust. Soc. Amer. 91*, 3540–3550.

Todd, N.P. 1995. The kinematics of musical expression. *J. Acoust. Soc. Amer. 97*, 1940–1949.

Toop, R. 1988. Four facets of the New Complexity. *CONTACT 32*, 4–50.

Widmer, G. 2000. Large-scale induction of expressive performance rules: First quantitative results. In

Proceedings of the 2000 International Computer Music Conference, Berlin, Germany, September 2000, I. Zannos, Ed. International Computer Music Association, San Francisco, CA, 344–347.

WIDMER, G. 2002. Machine discoveries: A few simple, robust local expression principles. *J. New Music Res. 31*, 37–50.

WIDMER, G. 2003. Discovering simple rules in complex data: A meta-learning algorithm and some surprising musical discoveries. *Artific. Intell. 146*, 129–148.

WIDMER, G. AND TOBUDIC, A. 2003. Playing Mozart by analogy: Learning multi-level timing and dynamics strategies. *J. New Music Res. 32*, 259–268.

WIDMER, G. 2004. Computational models of expressive music performance: The state of the art. *J. New Music Res. 33*, 203–216.

WITTEN, I. H. AND FRANK, E. 2000. *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*. Morgan Kaufmann, San Mateo, CA.

WRIGHT, M. AND BERDAHL, E. 2006. Towards machine learning of expressive microtiming in Brazilian drumming. In *Proceedings of the 2006 International Computer Music Conference*, New Orleans, USA, November 2006, I. Zannos, Ed. ICMA, San Francisco, CA, 572–575.

ZANON, P. AND DE POLI, G. 2003. Estimation of parameters in rule system for expressive rendering of musical performance. *Comput. Music J. 27*, 29–46.

ZANON, P. AND DE POLI, G. 2003. Time-varying estimation of parameters in rule systems for music performance. *J. New Music Res. 32*, 295–315.

ZHANG, Q. AND MIRANDA, E. R. 2006a. Evolving musical performance profiles using genetic algorithms with structural fitness. In *Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation*, (Seattle, Washington, July), J. V. Diggelen, M. A. Wiering, and E. D. D. Jong, Eds. ACM Press, New York, NY, 1833–1840.

ZHANG, Q. AND MIRANDA, E. R. 2006b. Towards an interaction and evolution model of expressive music performance. In *Proceedings of the 6th International Conference on Intelligent Systems Design and Applications* (Jinan, China, October), Y. Chen, and A. Abraham, Eds. IEEE Computer Society, Los Alamitos, CA, 1189–1194.

ZHANG, Q. AND MIRANDA, E. R. 2007. Evolving expressive music performance through interaction of artificial agent performers. In *Proceedings of the ECAL Workshop on Music and Artificial Life* (MusicAL, Lisbon, Portugal, September).

Received August 2007; revised February 2008, June 2008; accepted September 2008