Making Music with AI: Some examples

Ramón LOPEZ DE MANTARAS
IIIA-Artificial Intelligence Research Institute
CSIC-Spanish Scientific Research Council
Campus UAB 08193 Bellaterra

Abstract. The field of music raises very interesting challenges to computer science and in particular to Artificial Intelligence. Indeed, as we will see, computational models of music need to take into account important elements of advanced human problem solving capabilities such as knowledge representation, reasoning, and learning. In this paper I describe examples of computer programs capable of carrying out musical activities and describe some creative aspects of musical such programs.

Keywords. Artificial Intelligence, Computational Models of Music

Dedication to Rob Milne

Rob was a man that liked big challenges, especially in AI and in mountaineering, and therefore he was very enthusiastic about the challenge of replicating creativity by means of artificial intelligence techniques. In several occasions I had very long and stimulating discussions with him regarding artificial creativity in general and AI applications to music in particular, and he was well aware of the main developments in the area. This paper is dedicated to him, a truly creative man.

Introduction

Music is a very challenging application area for AI because, as we will see in this survey of a set of representative applications, it requires complex knowledge representation, reasoning, and learning. The survey is organized in three subsections. The first is devoted to compositional systems, the second describes improvisation systems, and the third is devoted to systems capable of generating expressive performances. It is unanimously accepted among researchers on AI and music that these three activities involve extensive creative processing. Therefore, although creativity is not the main focus of this paper, I believe that the computational systems described in this paper are valuable examples of artificially creative behaviour.

The books by Boden [1,2], Dartnall [3], Partridge & Rowe [4], and Bentley & Corne [5]; as well as the papers by Rowe & Partridge [6], and Buchanan [7] are very interesting sources of information regarding artificial intelligence approaches to creativity. Besides, for further information on AI and music I recommend the books edited by Balaban et al. [8] and by Miranda [9], and the book by Cope [10].

1. Composing music

Hiller and Isaacson's [11] work, on the ILLIAC computer, is the best known pioneering work in computer music. Their chief result is the "Illiac Suite", a string quartet composed following the "generate and test" problem solving approach. The program generated notes pseudo-randomly by means of Markov chains. The generated notes were next tested by means of heuristic compositional rules of classical harmony and counterpoint. Only the notes satisfying the rules were kept. If none of the generated notes satisfied the rules, a simple backtracking procedure was used to erase the entire composition up to that point, and a new cycle was started again. The goals of Hiller and Isaacson excluded anything related to expressiveness and emotional content. In an interview (see [11], p. 21), Hiller and Isaacson said that, before addressing the expressiveness issue, simpler problems needed to be handled first. We believe that this was a very correct observation in the fifties. After this seminal work, many other researchers based their computer compositions on Markov probability transitions but also with rather limited success judging from the standpoint of melodic quality. Indeed, methods relying too heavily on markovian processes are not informed enough to produce high quality music consistently.

However, not all the early work on composition relies on probabilistic approaches. A good example is the work of Moorer [13] on tonal melody generation. Moorer's program generated simple melodies, along with the underlying harmonic progressions, with simple internal repetition patterns of notes. This approach relies on simulating human composition processes using heuristic techniques rather than on Markovian probability chains. Levitt [14] also avoided the use of probabilities in the composition process. He argues that: "randomness tends to obscure rather than reveal the musical constraints needed to represent simple musical structures". His work is based on constraint-based descriptions of musical styles. He developed a description language that allows expressing musically meaningful transformations of inputs, such as chord progressions and melodic lines, through a series of constraint relationships that he calls "style templates". He applied this approach to describe a traditional jazz walking bass player simulation as well as a two-handed ragtime piano simulation.

The early systems by Hiller-Isaacson and Moore were both based also on heuristic approaches. However, possibly the most genuine example of early use of AI techniques is the work of Rader [15]. Rader used rule-based AI programming in his musical round (a circle canon such as "Frère Jacques") generator. The generation of the melody and the harmony were based on rules describing how notes or chords may be put together. The most interesting AI component of this system are the *applicability rules*, determining the applicability of the melody and chords generation rules, and the *weighting rules* indicating the likelihood of application of an applicable rule by means of a weight. We can already appreciate the use of metaknowledge in this early work.

AI pioneers such as Herbert Simon or Marvin Minsky also published works relevant to computer music. Simon and Sumner [16] describe a formal pattern language for music, as well as a pattern induction method, to discover patterns more or less implicit in musical works. One example of pattern that can be discovered is "the opening section is in C Major, it is followed by a section in dominant and then a return to the original key". Although the program was not completed, it is worth noticing that it was one of the firsts in dealing with the important issue of music modeling, a subject that has been, and still is, widely studied. For example, the use of models based on generative grammars has been, and continues to be, an important and very useful approach in music modeling (Lerdahl and Jackendoff [17])

Marvin Minsky in his well known paper *Music, Mind, and Meaning* [18] addresses the important question of how music impresses our minds. He applies his concepts of agent and its role in a society of agents as a possible approach to shed light on that question. For example, he hints that one agent might do nothing more than noticing that the music has a particular rhythm. Other agents might perceive small musical patterns such as repetitions of a pitch; differences such as the same sequence of notes played one fifth higher, etc. His approach also accounts for more complex relations within a musical piece by means of higher order agents capable of recognizing large sections of music. It is important to clarify that in that paper Minsky does not try to convince the reader about the question of the validity of his approach, he just hints at its plausibility.

Among the compositional systems there is a large number dealing with the problem of automatic harmonization using several AI techniques. One of the earliest works is that of Rothgeb [19]. He wrote a SNOBOL program to solve the problem of harmonizing the unfigured bass (given a sequence of bass notes infer the chords and voice leadings that accompany those bass notes) by means of a set of rules such as “If the bass of a triad descends a semitone, then the next bass note has a sixth”. The main goal of Rothgeb was not the automatic harmonization itself but to test the computational soundness of two bass harmonization theories from the eighteenth century.

One of the most complete works on harmonization is that of Ebcioglu [20]. He developed an expert system, CHORAL, to harmonize chorales in the style of J.S. Bach. CHORAL is given a melody and produces the corresponding harmonization using heuristic rules and constraints. The system was implemented using a logic programming language designed by the author. An important aspect of this work is the use of sets of logical primitives to represent the different viewpoints of the music (chords view, time-slice view, melodic view, etc.). This was done to tackle the problem of representing large amounts of complex musical knowledge.

MUSACT [21] uses Neural Networks to learn a model of musical harmony. It was designed to capture musical intuitions of harmonic qualities. For example, one of the qualities of a dominant chord is to create in the listener the expectancy that the tonic chord is about to be heard. The greater the expectancy, the greater the feeling of consonance of the tonic chord. Composers may choose to satisfy or violate these expectancies to varying degree. MUSACT is capable of learning such qualities and generate graded expectancies in a given harmonic context.

In HARMONET [22], the harmonization problem is approached using a combination of neural networks and constraint satisfaction techniques. The neural network learns what is known as harmonic functionality of the chords (chords can play the function of tonic, dominant, subdominant, etc) and constraints are used to fill the inner voices of the chords. The work on HARMONET was extended in the MELONET system [23, 24]. MELONET uses a neural network to learn and reproduce higher-level structure in melodic sequences. Given a melody, the system invents a baroque-style harmonization and variation of any chorale voice. According to the authors, HARMONET and MELONET together form a powerful music-composition system that generates variations whose quality is similar to those of an experienced human organist.

Pachet and Roy [25] also used constraint satisfaction techniques for harmonization. These techniques exploit the fact that both the melody and the harmonization knowledge impose constraints on the possible chords. Efficiency is however a problem with purely constraint satisfaction approaches.

Sabater et al. [26], approach the problem of harmonization using a combination of rules and case-based reasoning. This approach is based on the observation that purely rule-based harmonization usually fails because in general the rules don't make the music, it is the music that makes the rules. Then, instead of relying only on a set of imperfect rules, why not making use of the source of the rules, that is the compositions themselves? Case-based reasoning allows the use of examples of already harmonized compositions as cases for new harmonizations. The system harmonizes a given melody by first looking for similar, already harmonized, cases, when this fails, it looks for applicable general rules of harmony. If no rule is applicable, the system fails and backtracks to the previous decision point. The experiments have shown that the combination of rules and cases results in much fewer failures in finding an appropriate harmonization than using either technique alone. Another advantage of the case-based approach is that each newly correctly harmonized piece can be memorized and made available as a new example to harmonize other melodies; that is, a learning by experience process takes place. Indeed, the more examples the system has, the less often the system needs to resort to the rules and therefore it fails less. MUSE [27] is also a learning system that extends an initially small set of voice leading constraints by learning a set of rules of voice doubling and voice leading. It learns by reordering the rules agenda and by chunking the rules that satisfy the set of voice leading constraints. MUSE successfully learned some of the standard rules of voice leading included in traditional books of tonal music.

Certainly the best-known work on computer composition using AI is David Cope's EMI project [28, 29]. This work focuses on the emulation of styles of various composers. It has successfully composed music in the styles of Cope, Mozart, Palestrina, Albinoni, Brahms, Debussy, Bach, Rachmaninoff, Chopin, Stravinsky, and Bartok. It works by searching for recurrent patterns in several (at least two) works of a given composer. The discovered patterns are called signatures. Since signatures are location dependent, EMI uses one of the composer's works as a guide to fix them to their appropriate locations when composing a new piece. To compose the musical motives between signatures, EMI uses a compositional rule analyzer to discover the constraints used by the composer in his works. This analyzer counts musical events such as voice leading directions; use of repeated notes, etc. and represents them as a statistical model of the analyzed works. The program follows this model to compose the motives to be inserted in the empty spaces between signatures. To properly insert them, EMI has to deal with problems such as: linking initial and concluding parts of the signatures to the surrounding motives avoiding stylistic anomalies, maintaining voice motions, maintaining notes within a range, etc. Proper insertion is achieved by means of an Augmented Transition Network [30]. The results, although not perfect, are quite consistent with the style of the composer.

2. Synthesizing expressive performances

One of the main limitations of computer-generated music has been its lack of expressiveness, that is, lack of gesture. Gesture is what musicians call the nuances of performance that are unique (in the sense of conveying the "personal touch" of the musician) and subtly interpretive or, in other words, creative.

One of the first attempts to address expressiveness in music performances is that of Johnson [31]. She developed an expert system to determine the tempo and the articulation to be applied when playing Bach's fugues from "The Well-Tempered Clavier". The rules were obtained from two expert human performers. The output gives the base tempo value and a list of performance instructions on notes duration and articulation that should be followed by a human player. The results very much coincide with the instructions given in well known commented editions of "The Well-Tempered Clavier". The main limitation of this system is its lack of generality because it only works well for fugues written on a 4/4 meter. For different meters, the rules should be different. Another obvious consequence of this lack of generality is that the rules are only applicable to Bach fugues.

The work of Bresin, Friberg, Fryden, and Sundberg at KTH [32, 33, 34, 35] is one of the best known long term efforts on performance systems. Their current "Director Musices" system incorporates rules for tempo, dynamic, and articulation transformations constrained to MIDI. These rules are inferred both from theoretical musical knowledge and experimentally by training using, in particular, the so-called analysis-by-synthesis approach. The rules are divided in three main classes: *Differentiation rules*, which enhance the differences between scale tones; *Grouping rules*, which show what tones belong together; and *Ensemble rules*, that synchronize the various voices in an ensemble.

Canazza et al [36] developed a system to analyze how the musician's expressive intentions are reflected in the performance. The analysis reveals two different expressive dimensions: one related to the energy (dynamics) and the other one related to the kinetics (rubato) of the piece. The authors also developed a program for generating expressive performances according to these two dimensions.

The work of Dannenberg and Derenyi [37] is also a good example of articulation transformations using manually constructed rules. They developed a trumpet synthesizer that combines a physical model with a performance model. The goal of the performance model is to generate control information for the physical model by means of a collection of rules manually extracted from the analysis of a collection of controlled recordings of human performance.

Another approach taken for performing tempo and dynamics transformation is the use of neural network techniques. Bresin [38] describes a system that combines symbolic decision rules with neural networks is implemented for simulating the style of real piano performers. The outputs of the neural networks express time and loudness deviations. These neural networks extend the standard feed-forward network trained with the back propagation algorithm with feedback connections from the output neurons to the input neurons.

We can see that, except for the work done by the group at KTH that considers three expressive parameters, the other systems are limited to two such as rubato and dynamics, or rubato and articulation. This limitation has to do with the use of rules. Indeed, the main problem with the rule-based approaches is that it is very difficult to find rules general enough to capture the variety present in different performances of the same piece by the same musician and even the variety within a single performance [39]. Furthermore, the different expressive resources interact with each other. That is, the rules for dynamics alone change when rubato is also taken into account. Obviously, due to this interdependency, the more expressive resources one tries to model, the more difficult is finding the appropriate rules.

We have developed a system called SaxEx [40] SaxEx is a computer program capable of synthesizing high quality expressive tenor sax solo performances of jazz ballads based on cases representing human solo performances. Previous rule-based approaches to that problem could not deal with more than two expressive parameters (such as dynamics and rubato) because it is too difficult to find rules general enough to capture the variety present in expressive performances. Besides, the different expressive parameters interact with each other making it even more difficult to find appropriate rules taking into account these interactions.

With CBR, we have shown that it is possible to deal with the five most important expressive parameters: dynamics, rubato, vibrato, articulation, and attack of the notes. To do so, SaxEx uses a case memory containing examples of human performances, analyzed by means of spectral modeling techniques and background musical knowledge. The score of the piece to be performed is also provided to the system. The heart of the method is to analyze each input note determining (by means of the background musical knowledge) its role in the musical phrase it belongs to, identify and retrieve (from the case-base of human performances) notes with similar roles, and finally, transform the input note so that its expressive properties (dynamics, rubato, vibrato, articulation, and attack) match those of the most similar retrieved note. Each note in the case base is annotated with its role in the musical phrase it belong to as well as with its expressive values. Furthermore, cases do not contain just information on each single note but they include contextual knowledge at the phrase level. Therefore, cases in this system have a complex object-centered representation.

Although limited to monophonic performances, the results are very convincing and demonstrate that CBR is a very powerful methodology to directly use the knowledge of a human performer that is implicit in her playing examples rather than trying to make this knowledge explicit by means of rules. Some audio results can be listened to at www.iiia.csic.es/arcos/noos/Demos/Aff-Example.html. More recent papers by Arcos and Lopez de Mantaras [41] and by Lopez de Mantaras and Arcos [42], describe this system in great detail.

Based on the work on SaxEx, we have developed TempoExpress [43], a case-based reasoning system for applying musically acceptable tempo transformations to monophonic audio recordings of musical performances. TempoExpress has a rich description of the musical expressivity of the performances, that includes not only timing deviations of performed score notes, but also represents more rigorous kinds of expressivity such as note ornamentation, consolidation, and fragmentation. Within the tempo transformation process, the expressivity of the performance is adjusted in such a way that the result sounds natural for the new tempo. A case base of previously performed melodies is used to infer the appropriate expressivity. The problem of changing the tempo of a musical performance is not as trivial as it may seem because it involves a lot of musical knowledge and creative thinking. Indeed, when a musician performs a musical piece at different tempos the performances are not just time-scaled versions of each other (as if the same performance were played back at diferent speeds). Together with the changes of tempo, variations in musical expression are made (see for instance the work of Desain and Honing [44]). Such variations do not only affect the timing of the notes, but can also involve for example the addition or deletion of ornamentations, or the consolidation/fragmentation of notes. Apart from the tempo, other domain specific factors seem to play an important role in the way a melody is performed, such as meter, and phrase structure. Tempo transformation is one of the audio post-processing tasks manually done in audio-labs. Automatizing this process may, therefore, be of industrial interest.

Other applications of CBR to expressive performance are those of Suzuki et al. [45], and those of Tobudic and Widmer [46, 47]. Suzuki et al. [45], also use example cases of expressive performances to generate multiple performances of a given piece with varying musical expression, however they deal only with two expressive parameters. Tobudic and Widmer [46] apply instance-based learning (IBL) also to the problem of generating expressive performances. The IBL approach is used to complement a note-level rule-based model with some predictive capability at the higher level of musical phrasing. More concretely, the IBL component recognizes performance patterns, of a concert pianist, at the phrase level and learns how to apply them to new pieces by analogy. The approach produced some interesting results but, as the authors recognize, was not very convincing due to the limitation of using an attribute-value representation for the phrases. Such simple representation cannot take into account relevant structural information of the piece, both at the sub-phrase level and at the inter-phrasal level. In a subsequent paper, Tobudic and Widmer [47], succeeded in partly overcoming this limitations by using a relational phrase representation.

The possibility for a computer to play expressively is a fundamental component of the so-called "hyper-instruments". These are instruments designed to augment an instrument sound with such idiosyncratic nuances as to give it human expressiveness and a rich, live sound. To make an hyper-instrument, take a traditional instrument, like for example a cello, and connect it to a computer through electronic sensors in the neck and in the bow, equip also with sensors the hand that holds the bow and program the computer with a system similar to SaxEx that allows to analyze the way the human interprets the piece, based on the score, on musical knowledge and on the readings of the sensors. The results of such analysis allows the hyper-instrument to play an active role altering aspects such as timbre, tone, rhythm and phrasing as well as generating an accompanying voice. In other words, you have got an instrument that can be its own intelligent accompanist. Tod Machover, from MIT's Media Lab, developed a hypercello [48] and the great cello player Yo-Yo Ma premiered, playing the hypercello, a piece, composed by Tod Machover, called "Begin Again Again..." at the Tanglewood Festival several years ago.

3. Improvising music

Music improvisation is a very complex creative process that has also been computationally modeled. It is often referred to as “composition on the fly”. Because of the hard real time constraints involved, music improvisation it is creatively speaking more complex than composition (where musicians have the time to revise and improve their work) and since it obviously requires expressiveness too it is perhaps the most complex of the three music activities addressed in this paper. An early work on computer improvisation is the Flavours Band system of Fry [49]. Flavours Band is a procedural language, embedded in LISP, for specifying jazz and popular music styles. Its procedural representation allows the generation of scores in a pre-specified style by making changes to a score specification given as input. It allows combining random functions and musical constraints (chords, modes, etc.) to generate improvisational variations. The most remarkable result of Flavours Band was an interesting arrangement of the bass line, and an improvised solo, of John Coltrane’s composition *Giant Steps*.

GenJam [50] builds a model of a jazz musician learning to improvise by means of a genetic algorithm. A human listener plays the role of fitness function by rating the offspring improvisations. Papadopoulos and Wiggins [51] also used a genetic algorithm to improvise jazz melodies on a given chord progression. Contrarily to GenJam, the program includes a fitness function that automatically evaluates the quality of the offspring improvisations rating eight different aspects of the improvised melody such as the melodic contour, notes duration, intervallic distances between notes, etc.

Franklin [52] uses recurrent neural networks to learn how to improvise jazz solos from transcriptions of solo improvisations by saxophonist Sonny Rollins. A reinforcement learning algorithm is used to refine the behavior of the neural network. The reward function rates the system solos in terms of jazz harmony criteria and according to Rollins style.

The lack of interactivity, with a human improviser, of the above approaches has been criticized [53] on the grounds that they remove the musician from the physical and spontaneous creation of a melody. Although it is true that the most fundamental characteristic of improvisation is the spontaneous, real-time creation of a melody, it is also true that interactivity was not intended in these approaches and nevertheless they could generate very interesting improvisations. Thom [53] with her Band-out-of-a-Box (BoB) system addresses the problem of real-time interactive improvisation between BoB and a human player. In other words, BoB is a "music companion" for real-time improvisation. Thom's approach follows Johnson-Laird's [54] psychological theory of jazz improvisation. This theory opposes the view that improvising consists of rearranging and transforming pre-memorized "licks" under the constraints of a harmony. Instead he proposes a stochastic model based on a greedy search over a constrained space of possible notes to play at a given point in time. The very important contribution of Thom is that her system learns these constraints, and therefore the stochastic model, from the human player by means of an unsupervised probabilistic clustering algorithm. The learned model is used to abstract solos into user-specific playing modes. The parameters of that learned model are then incorporated into a stochastic process that generates the solos in response to four bar solos of the human improviser. BoB has been very successfully evaluated by testing its real-time solo tradings in two different styles, that of saxophonist Charlie Parker, and that of violinist Stephane Grapelli.

Another remarkable interactive improvisation system was developed by Dannenberg [55]. The difference with Thom's approach is that in Dannenberg's system, music generation is mainly driven by the composer's goals rather than the performer's goals. Wessel's [56] interactive improvisation system is closer to Thom's in that it also emphasizes the accompaniment and enhancement of live improvisations.

A very recent and very remarkable interactive musical system is that of Pachet [57]. His system, Continuator, is based on extended multilayer Markov models to learn to interactively play with a user in the users' style and therefore it allows to carry musical dialogues between a human and the system.

4. Apparently or really creative

The described computational approaches to composing, performing, and improvising music are not just successful examples of AI applications to music. In my opinion are also valid examples of artificially creative systems because composing, performing, and improvising music are, undoubtedly, highly creative activities. Margaret Boden pointed out that even if an artificially intelligent computer would be as creative as Bach or Einstein, for many it would be just apparently creative but not really creative. I fully agree with Margaret Boden in the two main reasons for such rejection. These reasons are: the lack of intentionality and our reluctance to give a place in our society to artificially intelligent agents. The lack of intentionality is a direct consequence of Searle's Chinese room argument, which states that computer programs can only perform syntactic manipulation of symbols but are unable to give them any semantics. This critic is based on an erroneous concept of what a computer program is. Indeed, a computer program does not only manipulate symbols but also triggers a chain of cause-effect relations inside the computer hardware and this fact is relevant for intentionality since it is generally admitted that intentionality can be explained in terms of causal relations. However, it is also true that existing computer programs lack too many relevant causal connections to exhibit intentionality but perhaps future, possibly anthropomorphic, embodied artificial intelligences, that is agents equipped not only with sophisticated software but also with different types of advanced sensors allowing to interact with the environment, may have enough causal connections to have intentionality.

Regarding social rejection, the reasons why we are so reluctant to accept that non human agents can be creative is that they do not have a natural place in our society of human beings and a decision to accept them would have important social implications. It is therefore much simpler to say that they appear to be intelligent, creative, etc. instead of saying that they are. In a word, it is a moral but not a scientific issue. A third reason for denying creativity to computer programs is that they are not conscious of their accomplishments. However I agree with many AI scientists in thinking that the lack of consciousness is not a fundamental reason to deny the potential for creativity or even the potential for intelligence. After all, computers would not be the first example of unconscious creators, evolution is the first example as Stephen Jay Gould [58] brilliantly points out: *If creation demands a visionary creator, then how does blind evolution manage to build such splendid new things as ourselves?*

References

[1] M. Boden, *The Creative Mind: Myths and Mechanisms*, Basic Books 1991.

[2] M. Boden (Ed.); *Dimensions of Creativity*, MIT Press 1994.

[3] T. Dartnall (Ed.); *Artificial Intelligence and Creativity*, Kluwer Academic Pub. 1994.

[4] D. Partridge, and J. Rowe; *Computers and Creativity*, Intellect Books, 1994.

[5] P.J. Bentley and D.W: Corne (eds.), *Creative Evolutionary Systems*, Morgan Kaufmann 2002.

[6] J. Rowe, D. Partridge, Creativity: A survey of AI approaches, *Artificial Intelligence Review* 7 (1993), 43-70.

[7] B.G. Buchanan, Creativity at the Metalevel: AAAI-2000 Presidential Address, *AI Magazine* 22:3 (2001), 13-28.

[8] M. Balaban, K. Ebcioglu, and O. Laske (Eds.); *Understanding Music with AI*, MIT Press1992.

[9] E. R. Miranda (Ed.); Readings in Music and Artificial Intelligence, Harwood Academic Pub. 2000.

[10] D. Cope, *The Algorithmic Composer*, A-R Editions, Computer Music and Audio Digital Series, Volume 16, 2000.

[11] L. Hiller, and L. Isaacson. *Musical composition with a high-speed digital computer* (1958). Reprinted in Schwanauer, S.M., and Levitt, D.A., ed. *Machine Models of Music*. 9-21. Cambridge, Mass.: The MIT Press, 1993.

[12] S.M. Schwanauer, and D.A. Levitt, D.A., ed. *Machine Models of Music*. Cambridge, Mass.: The MIT Press, 1993.

[13] J.A. Moorer. *Music and computer composition* (1972). Reprinted in Schwanauer, S.M., and Levitt, D.A., ed. *Machine Models of Music*. 167-186. Cambridge, Mass.: The MIT Press. 1993.

[14] D.A. Levitt. *A Representation for Musical Dialects*. In *Machine Models of Music*, ed. Schwanauer, S.M., and Levitt, D.A., 455-469. Cambridge, Mass.: The MIT Press. 1993.

[15] G.M. Rader. *A method for composing simple traditional music by computer* (1974). Reprinted in Schwanauer, S.M., and Levitt D.A., ed.. *Machine Models of Music*. 243-260. Cambridge, Mass.: The MIT Press. 1993.

[16] H.A. Simon, and R.K. Sumner. *Patterns in Music* (1968). Reprinted in Schwanauer, S.M., and Levitt, D.A., ed. *Machine Models of Music*. 83-110. Cambridge, Mass.: The MIT Press. 1993.

[17] F. Lerdahl, and R. Jackendoff. *A Generative Theory of Tonal Music*, MIT Press 1983.

[18] M. Minsky. *Music, Mind, and Meaning.* (1981). Reprinted in Schwanauer, S.M., and Levitt, D.A., ed. *Machine Models of Music*. 327-354. Cambridge, Mass.: The MIT Press. 1993.

[19] J. Rothgeb. *Simulating musical skills by digital computer* (1969). Reprinted in Schwanauer, S.M., and Levitt, D.A., ed. *Machine Models of Music*. 157-164. Cambridge, Mass.: The MIT Press. 1993.

[20] K. Ebcioglu. *An expert system for harmonizing four-part chorales*. In *Machine Models of Music*, ed. Schwanauer, S.M., and Levitt, D.A., 385-401. Cambridge, Mass.: The MIT Press. 1993.

[21] J. Bharucha. *MUSACT: A connectionist model of musical harmony*. In *Machine Models of Music*, ed. Schwanauer, S.M., and Levitt, D.A., 497-509. Cambridge, Mass.: The MIT Press. 1993.

[22] J. Feulner. *Neural Networks that learn and reproduce various styles of harmonization.* In Proceedings of the 1993 International Computer Music Conference. San Francisco: International Computer Music Association. 1993.

[23] D. Hörnel, and P. Degenhardt. *A neural organist improvising Baroque-style melodic variations*. In Proceedings of the 1997 International Computer Music Conference, 430-433. San Francisco: International Computer Music Association. 1997.

[24] D. Hörnel, and W. Menzel. Learning musical strucure and style with Neural Networks. *Journal on New Music Research*, 22:4 (1998), 44-62.

[25] F. Pachet, and P. Roy. *Formulating constraint satisfaction problems on part-whole relations: The case of automatic harmonization*. In ECAI'98 Workshop on Constraint Techniques for Artistic Applications. Brighton, UK. 1998.

[26] J. Sabater, J.L. Arcos, and R. Lopez de Mantaras. *Using Rules to Support Case-Based Reasoning for Harmonizing Melodies*. In AAAI Spring Symposium on Multimodal Reasoning, pp. 147-151. Menlo Park, Calif.: American Association for Artificial Intelligence. 1998.

[27] S.M. Schwanauer. *A learning machine for tonal composition*. In *Machine Models of Music*, ed. Schwanauer, S.M., and Levitt, D.A., 511-532. Cambridge, Mass.: The MIT Press. 1993.

[28] D. Cope. *Experiments in Music Intelligence*. A-R Editions 1996.

[29] D. Cope. *Pattern Matching as an engine for the computer simulation of musical style*. In Proceedings of the 1990 International Computer Music Conference. San Francisco, Calif.: International Computer Music Association. 1990.

[30] W. Woods. Transition network grammars for natural language analysis. *Communications of the ACM* 13:10 (1970), 591-606.

[31] M.L. Johnson. *An expert system for the articulation of Bach fugue melodies*. In *Readings in Computer Generated Music*, ed. D.L. Baggi, 41-51. Los Alamitos, Calif.: IEEE Press. 1992.

[32] R. Bresin. *Articulation rules for automatic music performance*. In Proceedings of the 2001 International Computer Music Conference 2001. San Francisco, Calif.: International Computer Music Association. 2001.

[33] A. Friberg. *A quantitative rule system for musical performance*. PhD dissertation, KTH, Stockholm. 1995.

[34] A. Friberg, R. Bresin, L. Fryden, and J. Sunberg. Musical punctuation on the microlevel: automatic identification and performance of small melodic units. *Journal of New Music Research* 27:3 (1998), 271-292.

[35] A. Friberg, J. Sunberg, and L. Fryden. Music From Motion: Sound Level Envelopes of Tones Expressing Human Locomotion. *Journal on New Music Research*, 29:3 (2000), 199-210.

[36] S. Canazza, G. De Poli, A. Roda, and A. Vidolin. *Analysis and synthesis of expressive intention in a clarinet performance*. In Proceedings of the 1997 International Computer Music Conference, 113-120. San Francisco, Calif.: International Computer Music Association. 1997.

[37] R.B. Dannenberg, and I. Derenyi. Combining instrument and performance models for high quality music synthesis, *Journal of New Music Research* 27:3 (1998), 211-238.

[38] R. Bresin. Artificial neural networks based models for automatic performance of musical scores, *Journal of New Music Research* 27:3 (1998), 239-270.

[39] R.A. Kendall, and E.C. Carterette. The communication of musical expression. *Music Perception* 8:2 (1990), 129.

[40] J.L. Arcos, R. Lopez de Mantaras, and X. Serra; Saxex: A Case-Based Reasoning System for Generating Expressive Musical Performances. *Journal of New Music Research* 27:3 (1998), 194-210.

[41] J.L. Arcos, and R. Lopez de Mantaras; An Interactive Case-Based Reasoning Approach for Generating Expressive Music. *Applied Intelligence* 14:1 (2001), 115-129.

[42] R. Lopez de Mantaras, and J.L. Arcos; AI and Music: From Composition to Expressive Performance. *AI Magazine* 23:3 (2002), 43-57.

[43] M. Grachten, J.L. Arcos, and R. Lopez de Mantaras; *TempoExpress, a CBR Approach to Musical Tempo Transformations*. In Proceedings of the 7th European Conference on Case-Based Reasoning (Eds. P. Funk and P. A. Gonzalez Calero), Lecture Notes in Artificial Intelligence 3155 (2004), 601-615.

[44] P. Desain and H. Honing. *Tempo curves considered harmful.* In ”Time in contemporary musical thought” J. D. Kramer (ed.), Contemporary Music Review. 7:2, 1993.

[45] T. Suzuki, T. Tokunaga, and H. Tanaka; *A Case-Based Approach to the Generation of Musical Expression*. In Proceedings of the 16th International Joint Conference on Artificial Intelligence, Morgan Kaufmann (1999), 642-648.

[46] A. Tobudic, and G. Widmer; *Playing Mozart Phrase by Phrase*. In Proceedings of the 5th International Conference on Case-Based Reasoning (Eds. K.D. Ashley and D.G. Bridge), Lecture Notes in Artificial Intelligence 3155 (2003), 552-566.

[47] A. Tobudic, and G. Widmer; *Case-Based Relational Learning of Expressive Phrasing in Classical Music*. In Proceedings of the 7th European Conference on Case-Based Reasoning (Eds. P. Funk and P. A. Gonzalez Calero), Lecture Notes in Artificial Intelligence 3155 (2004), 419-433.

[48] T. Machover, *Hyperinstruments: A Progress Report*, MIT Media Lab Internal Research Report, 1992.

[49] C. Fry. *Flavors Band: A language for specifying musical style* (1984). Reprinted in Schwanauer, S.M., and Levitt, D.A., ed. 1993. *Machine Models of Music*. 427-451. Cambridge, Mass.: The MIT Press.

[50] J.A. Biles. GenJam: *A genetic algorithm for generating Jazz solos.* In Proceedings of the 1994 International Computer Music Conference. San Francisco, Calif.: International Computer Music Association. 1994.

[51] G. Papadopoulos, and G. Wiggins. *A genetic algorithm for the generation of Jazz melodies.* In Proceedings of the SteP'98 Conference, Finland. 1998

[52] J.A. Franklin. *Multi-phase learning for jazz improvisation and interaction*. In Proceedings of the Eight Biennial Symposium on Art and Technology. 2001.

[53] B. Thom. *BoB*: *An improvisational music companion*. PhD dissertation, School of Computer Science, Carnegie-Mellon University. 2001.

[54] P.N. Johnson-Laird. *Jazz improvisation: A theory at the computational level.* In *Representing Musical Structure*, ed. P. Howell, R. West, and I. Cross. London, UK and San Diego, Calif.: Academic Press. 1991.

[55] R.B. Dannenberg. Software design for interactive multimedia performance. *Interface* 22:3 (1993), 213-218.

[56] D. Wessel, M. Wright, and S.A. Kahn. *Preparation for improvised performance in collaboration with a Khyal singer*. In Proceedings of the 1998 International Computer Music Conference. San Francisco, Calif.: International Computer Music Association. 1998.

[57] François Pachet: Beyond the Cybernetic Jam Fantasy: The Continuator. *IEEE Computer Graphics and Applications* 24:1 (2004), 31-35.

[58] S. J. Gould; Creating the Creators, *Discover Magazine*, October (1996), 42-54.