

# Simulating Creativity in Jazz Performance

Geber Ramalho

Jean-Gabriel Ganascia

LAFORIA-IBP-CNRS

Université Paris VI - 4, Place Jussieu

75252 Paris Cedex 05 - FRANCE

Tel. (33-1) 44.27.37.27 - Fax. (33-1) 44.27.70.00

e-mails: {ramalho, ganascia}@laforia.ibp.fr

## Abstract

This paper considers the problem of simulating creativity in the domain of Jazz improvisation and accompaniment. Unlike most current approaches, we try to model the musicians' behavior by taking into account their experience and how they use it with respect to the evolving contexts of live performance. To represent this experience we introduce the notion of *Musical Memory*, which explores the principles of Case-Based Reasoning (Slade 1991). To produce live music using this *Musical Memory* we propose a problem solving method based on the notion of PACTs (*Potential ACTions*) that are activated according to the context and then combined in order to produce notes. We show that our model supports two of the main features of creativity: non-determinism and absence of well-defined goals (Johnson-Laird 1992).

## 1 - Introduction

Our research is concerned with the study of the strengths and limitations of AI techniques to simulate creative behavior on a computer. Although creativity has always been present on the AI research agenda there is no accurate understanding of human creativity; its simulation on a computer still remains an open problem (AAAI 1993). In fact, there is an apparent paradox in the formalization of creativity due to the common sense opinion that, by definition, creativity embodies what cannot be formalized.

To avoid both real and imaginary difficulties of simulating creative behavior on a computer, we have decided to concentrate on modeling particular kinds of creative activities such as musical ones. We do not intend to model creativity from a psychological point of view but rather to investigate it by seeking the simple computational mechanisms that may underlie it. In other words, we attempt to model creativity in terms of problem solving (Newell & Simon 1972; Nilsson 1971, Laird, Newell & Rosembloom, 1987).

We have chosen to work on Jazz improvisation and accompaniment because of their spontaneity, in contrast to the formal aesthetic of contemporary classical music composition. From an AI point of view, modeling Jazz performance raises interesting problems since performance requires both theoretical knowledge and great

skill. In addition, Jazz musicians are encouraged to develop their musical abilities by listening and practicing rather than studying in *conservatoires* (Baker 1980).

In Section 2 we present briefly the problems of modeling musical creativity in Jazz performance. We show the relevance of taking into account the fact that musicians integrate rules and memories dynamically according to the context. In Section 3 we introduce two basic notions of our model: PACTs and Musical Memory. A general description of our model is given in Section 4. In Section 5 we give further details about the modules of our model, showing particularly how the composition module integrates the two above-mentioned notions to create music. In the last section we discuss our current work and directions for further developments.

## 2 - Modeling Musical Creativity

### 2.1 - The Problem and the Current Approaches

Let us begin by defining some simple musical concepts. A note is a triplet (pitch, duration, amplitude) and can be considered as the basic tonal music element. Putting notes together one obtains other musical elements such as a melody (temporal sequence of notes) or a chord (set of simultaneous notes). Scales and rhythm concern respectively the pitches and durations of a set of notes. The tasks of improvisation and accompaniment consist in playing notes (melodies and/or chords) according to the guidelines laid down in a given chord grid (sequence of chords underlying the song). But, it is in the strikingly large gap between the actually played music and the chord grid instructions that the richness of live Jazz performance lies (Ramalho & Pachet 1994).

Musicians cannot justify all the local choices they make (typically at note-level) even if they have consciously applied some strategies in the performance. This is the greatest problem of modeling the knowledge used to fill the gap referred to above. To face this problem, the first approach is to make random-oriented choices from a library of musical patterns weighted according to their frequency of use (Ames & Domino 1992). The second approach focuses on very detailed descriptions so as to obtain a complete explanation of musical choices in terms of rules or grammars (Steedman

1984). In the first case, since there is no explicit semantics associated to random-oriented choices, it is difficult to control changes at more abstract levels than the note level. In the second, the determinism of rule-based framework lacks flexibility because of the introduction of "artificial" or over-specialized rules that do not correspond to the actual knowledge used by musicians. This crucial trade-off between "flexibility and randomness" and "control and semantics" affects the modeling of other creative activities too (Rowe & Partridge 1993).

## 2.2 - Claims on Knowledge and Reasoning in Jazz Performance

If musical creativity is neither a random activity nor a fully explainable one, then creativity modeling requires a deeper understanding of the nature and use of musical knowledge. This section presents two general results of our early work where we interviewed Jazz musicians and recorded live performances in order to elicit this knowledge.

Our first claim is that Jazz musicians' activities are supported by two main knowledge structures: memories and rules. More specifically, we claim that these memories are the main source of knowledge in intuitive composition tasks and that most Jazz rules are either abstract or incomplete with respect to their possibility of directly determining the notes to be played. Jazz musicians use rules they have learned in schools and through Jazz methods (Baudoin 1990). However, these rules do not embody all knowledge. For example, there is no logical rule chaining that can directly instantiate important concepts such as tension, style, swing and contrast, in terms of notes. This phenomenon is a consequence of the Jazz learning process which involves listening to and imitating performances of great Jazz stars (Baker 1980). The experience thus acquired seems to be stored in a long term musical memory.

To put it in a nutshell, musicians integrate rules and memories into their actions dynamically. Sometimes, note-level rules (that determine the notes directly) are applied but, very often, these rules are not available. In these cases a fast search for appropriate musical fragments in the musician's auditory memory is carried out using the available general rules. This memory search is both flexible and controlled because of the mechanism of partial matching between the memory contents and requirements stated by the general rules. In terms of modeling, this is an alternative approach that avoids the need for "artificial" rules or randomness.

Our second claim is that musical actions depend strongly on contexts that evolve over time. The great interaction between either musicians themselves or musicians and the public/environment may lead them to reinforce or discard their initial strategies while performing. The constraints imposed by real-time performance force musicians to express their knowledge

as a fast response to on-going events rather than as an accurate search for "the best musical response". Jazz creativity occurs within the continuous confrontation between the musician's background knowledge and the context of live performance.

## 3- Two Basic Notions of our Model

### 3.1 - Potential ACTions (PACTs)

Pachet (Pachet 1990) has proposed the notion of PACTs (at this time called "strategies") as a generic framework for representing the potential actions (or intentions) that musicians may take within the context of performance. Focusing the modeling on musical actions rather than on the syntactic dimension of notes, additional knowledge can be expressed. In fact, PACTs can represent not only notes but also incomplete and abstract actions, as well as action chaining. PACTs are frame-like structures whose *main attributes* are: start-beat, end-beat, dimensions, abstract-level, type and instrument-dependency. Let us now see how PACTs are described, through a couple of examples.

PACTs are activated at a precise moment in time and are of limited duration which can correspond to a group of notes, a chord, a bar, the entire song, etc. PACTs may rely on different dimensions of notes: rhythm (r); amplitude (a); pitch (p) and their arrangements (r-a, r-p, p-a, r-p-a). When its dimensions are instantiated, the abstract level of a PACT is *low*, otherwise it is *high*. For instance, "play loud", "play this rhythm" and "play an ascending arpeggio" are low-level PACTs on amplitudes, rhythm and pitches respectively. "Play this lick transposed one step higher" is a low-level PACT on all three dimensions. "Play syncopated" and "use major scale" are high-level on respectively rhythm and pitches. PACTs can be of two types: *procedural* (e.g. "play this lick transposed one step higher") or *property-setting* (e.g. "play bluesy"). PACTs may also depend on the instrument. For example, "play five-note chord" is a piano PACT whereas "play stepwise" is a bass PACT.

For the sake of simplicity we have not presented many other descriptors that are needed according to the nature and abstract level of the PACTs. For instance, pitch PACTs have descriptors such as pitch-contour (ascending, descending, etc.), pitch-tessitura (high, low, middle, etc.), pitch-set (triad, major scale, dorian mode, etc.) and pitch-style (dissonant, chord-based, etc.).

From the above description two important properties of PACTs appear. The first one is the *playability* of a PACT. The less abstract a PACT is and the more dimensions it relies on, the more it is "playable" (e.g. "play ascending notes" is less playable than "play C E G", "play bluesy" is less playable than "play a diminished fifth on the second beat", etc.). A *fully playable* (or just *playable*) PACT is defined as a low-level PACT on all three dimensions. The second property is the *combinability* of PACTs, i.e. they can be combined

to generate more playable PACTs. For instance, the PACT “play ascending notes” may combine with “play triad notes” in a given context (e.g. C major) to yield “play C E G”. In this sense, PACTs may or may not be compatible. “Play loudly” and “play quietly” cannot be combined whereas “swing”, “play major scale” and “play loudly” can. These properties constitute the basis of our problem solving method. As discussed in Section 4, solving a musical problem consists in assembling (combining) a set of PACTs that have been activated by the performance context.

### 3.2 - Musical Memory

There is no guarantee that a given set of PACTs contains the necessary information so as to produce a playable PACT. As discussed in Section 2.2, this lack of information is related to the fact that musical choices cannot be fully expressed in terms of logical rule chaining, i.e. *Jazz rules* are often either abstract or incomplete to determine directly the notes to be played. To solve this problem we have introduced the notion of Musical Memory which explores the principles of case-based reasoning [Slade 91]. This Musical Memory is a long term memory that accumulates the musical material (cases) the musicians have listened to. These cases can be retrieved and modified to provide missing information.

The contents and representation of the Musical Memory can be determined: the cases must correspond to low-level PACTs that can be retrieved during the problem solving according to the information contained in the activated PACTs. These cases are obtained by applying transformations (e.g. time segmentation, projection on one or two dimensions, etc.) to transcriptions of actual Jazz recordings. This process (so far, guided by a human expert) yields cases such as melody fragments, rhythm patterns, amplitude contours, chords, etc. The cases are indexed from various points of view that can have different levels of abstraction such as underlying chords, position within the song, amplitude, rhythmic and melodic features (Ramalho & Ganascia 94). These features are in fact the same ones used to describe high-level PACTs. For instance, pitches are described in terms of contour, tessitura, set and style as discussed in last section.

It is important to stress that high-level PACTs have also been determined from transcriptions of Jazz recordings but not automatically, since this would require much more complex transformations on the transcriptions. These PACTs were in fact acquired during an earlier knowledge acquisition phase working with experts.

## 4 - General Description of our Model

### 4.1 - What is a Musical Problem?

Johnson-Laird (Johnson-Laird 1992) among other

researchers has identified three features of creative tasks that show the difficulties of formalizing creativity as classical problem solving (Newell & Simon 1972; Nilsson 1971): non-determinism (for the same given composition problem it is possible to obtain different musical solutions which are all acceptable); absence of well-defined goals (there is only a vague impression of what is to be accomplished, i.e. goals are refined or changed in the on-going process); no clear point of termination (because of both the absence of a clear goal and the absence of aesthetic consensus for evaluating results).

Taking an initial state of a problem space as a time segment (e.g. bars) with no notes, a musical problem consists in filling this time segment with notes which satisfy some criteria. This intuitive formulation of what a musical problem is underlies the above criticism of formalizing musical creativity. Some AI researchers have encountered many difficulties in exploring this point of view (see for instance Vicinanza's work (Vicinanza & Prietula 1989) on generating tonal melodies). However, we present here a different point of view that allows us to formalize and deal with musical creativity as problem solving. We claim that the musical problem is in fact to know how to start from a “vague impression” and go towards a precise specification of these criteria. In other words, the initial state of the music problem space could be any set of PACTs within a time interval and the goal could be a unique playable PACT. The goal is fixed and clearly defined (i.e. the goal is to play!) and solving the problem is equivalent to assembling or combining PACTs. An associated musical problem would be to determine the time interval continuously so as to reach the end of the song.

### 4.2 - The Reasoner

What we do is model a musician as a *reasoner* whose behavior is simulated by three modules which work coordinately in parallel (see Figure 1). The modules of our model resemble the *Monitoring*, *Planning* and *Executing* ones of some robotics applications (Ambros-Ingerson & Steel 1988). The *context* is composed of a *chord grid* which is given at the outset and *events* that occur as the performance goes on, i.e. the notes played by the orchestra and reasoner and also the public reactions. The *perception module* “listens to” the context events and puts them in the *Short-Term Memory*. The *composing module* computes the notes (a playable PACT) which will be executed in the future time segment of the chord grid. This is done using three elements: the Short-Term Memory contents, the reasoner's mood and the chords of the future chord grid segment. The *reasoner's Mood* changes according to the context events. The *execution module* works on the current chord grid segment by executing the playable PACT previously provided by the composing module. This execution corresponds to the sending of note

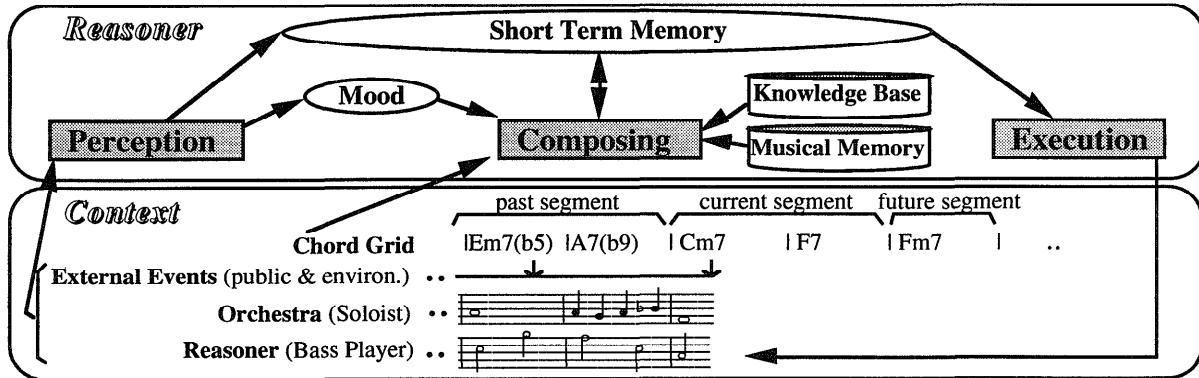


Figure 1 - Overall Description of the Model

information at their start time to the perception module and to a MIDI synthesizer, which generates the corresponding sound.

## 5 - Components of our Model

### 5.1 - The Perception Module

Modeling the dialog between musicians and their interaction with the external environment is a complex problem since the context events are unpredictable and understanding them depends on cultural and perceptual considerations.

To achieve an initial validation of our model, our current work focuses on the implementation of the composing module, since it is at the heart of the improvisation tasks. And instead of implementing the perception module, we have proposed a structure called a Performance Scenario which is a simpler yet still powerful representation of the evolving context. The idea is to control the context events by asking for the user's aid. Before the performance starts, the user imagines a virtual external environment and characterizes it by choosing some features and events from a limited repertoire and assigning an occurrence time to the events. As for the dialog between the musicians, the user listens to a previous orchestra recording and gives a first level interpretation by leaving some marks such as "soloist using dorian mode in a cool atmosphere" or "soloist is playing this riff". In short, the Performance Scenario is composed of marks that are obtained from the interpretation of the orchestra part and the setting of external environment events. These marks are only available to the system at their specified start time.

Unfortunately, the user cannot interpret the notes the reasoner himself has just played. However, the reasoner can take into account some simple features of these notes (e.g. last note, pitch and amplitude direction, etc.) when activating and assembling PACTs.

### 5.2 - The Composition module

The problem of playing along a given chord grid can be viewed as a continuous succession of three sub-problems: establishing the duration of the new chord grid segment; determining the PACTs associated to this segment; and assembling this group of PACTs in order to generate a unique playable PACT. The first two are more questions of problem setting, the third is a matter of problem solving and planning.

The composition model is supported by a Musical Memory and Knowledge Base. The former contains low-level PACTs that can be retrieved during the PACT assembly. The latter contains production rules and heuristics concerned with the segmentation of the chord grid, changes in the Mood and the selection/activation of PACTs. These rules are also used to detect and solve incompatibilities between PACTs, to combine PACTs and to modify low-level PACTs retrieved from the Musical Memory.

#### 5.2.1 - Segmenting the Chord Grid and Selecting PACTs

The chord grid is segmented in non regular time intervals corresponding to typical chord sequences (II-V cadences, modulations, turnarounds, etc.) abundantly catalogued in Jazz literature (Baudoin 1990). In fact, the reasoning of musicians does not progress note by note but by "chunks" of notes (Sloboda 1985). The criteria for segmenting the chord grid are simple and are the same as those used for segmenting the transcription of Jazz recordings in order to build the Musical Memory.

Given the chord grid segment, the group of associated PACTs derives from three sources. Firstly, PACTs are activated according to the chords of the grid segment (e.g. "if two chords have a long duration and a small interval distance between them then play an ascending arpeggio"). Other PACTs are activated from the last context events (e.g. "if soloist goes in descending direction then follow him"). The activation of a PACT corresponds to the assignment of values to its attributes, i.e. the generation of an instance of the class PACT in an Object-Oriented

Language. Finally, the previously activated PACTs whose life time intersects the time interval defined by the segmentation (e.g. “during the improvisation play louder”) are added to the group of PACTs obtained from the first two steps.

The reasoner can be seen as an automaton whose state (Mood) changes according to the context events (e.g. “if no applause after solo then Mood is *bluesy*” or “if planning is late with respect to the execution then Mood is *in a hurry*”). So far, the reasoner’s Mood is characterized by a simple set of “emotions”. In spite of its simplicity, the Mood plays a very important role in the activation and assembling of PACTs. It appears in the left-hand side of some rules for activating PACTs and also has an influence on the heuristics that establish the choice preferences for the PACT assembly operators. For instance, when the reasoner is “*in a hurry*” some incoming context events may not be considered and the planning phase can be bypassed by the activation of playable PACTs (such as “play this lick”) which correspond to the various “default solutions” musicians play.

### 5.2.2 - Assembling PACTs

The initial state of the assembly problem space is a group of selected PACTs corresponding to the future chord grid segment. The goal is a playable PACT. A new state can be reached by the application of three operators or operator schemata (since they must previously have been instantiated to be applied): delete, combine and add. The choice of operator follows an opportunistic problem solving strategy which seeks the shortest way to reach the goal. Assembling PACTs is a kind of planning whose *space state* is composed of *potential actions* that are combined both in parallel and sequentially since sometimes they may be seen as constraints and other times as procedures. Furthermore, the actions are not restricted to primary ones since potential actions have different abstract levels. Finally, there is no backtracking in the operator applications.

The *delete operator* is used to solve conflicts between PACTs by eliminating some of them from the group of PACTs that constitute the next state of the space problem. For instance, the first two of the PACTs “play ascending arpeggio”, “play in descending direction”, “play louder” and “play syncopated” are incompatible. As proposed in SOAR (Laird, Newell & Rosenbloom, 1987), heuristics state the preferences for choosing a production rule from a set of fireable rules. In our example, we eliminate the second one because the first one is more playable.

The *combine operator* transforms compatible PACTs into a new one. Sometimes the information contained in the PACTs can be merged immediately to yield a low-level PACT on one or more dimensions (e.g. “play ascending notes” with “play triad notes” yields “play C E G” in a C major context). Other times, the information is only placed side by side in the new PACT waiting for

future merger (e.g. “play louder” and “play syncopated” yields, say, “play louder and syncopated”). Combining this with “play ascending arpeggio” generates a playable PACT.

The *add operator* supplies the missing information that is necessary to assemble a playable PACT by retrieving and adapting adequate cases (low-level PACTs on one or more dimensions) from the Musical Memory. The retrieval is done by a partial pattern matching between case indexes, the chords of the chord grid segment and the current activated PACTs. Since the concepts used in the indexation of cases correspond to the descriptors of high-level PACTs, it is possible to retrieve low-level PACTs when only high-level PACTs are activated. For instance, if the PACTs “play bluesy” and “play a lot of notes” are activated in the context of “Bb7-F7” chords, we search for a case that has been indexed as having a bluesy style, a lot of notes and IV7-I7 as underlying chords. When there is no PACT on a particular dimension, we search for a case that has “default” as a descriptor of this dimension. For instance, it is possible to retrieve a melody even when the activated PACTs concern amplitudes only.

The cases may correspond to some “chunks” of the note dimensions that may not *fit in* the “gaps” that exist in the current activated PACTs. Thus, retrieved cases may carry additional information which can be partially incompatible with the activated PACTs. Here either the conflicting information is ignored or it can “short-circuit” the current PACT assembly and lead to a different playable PACT. Let us suppose that the activated PACTs concern pitches and amplitudes and the retrieved case concerns pitches and rhythm. Only the activated PACTs on amplitude can be considered to be combined with the retrieved case generating a playable PACT. But, if the retrieved case concerns rhythm and amplitudes, perhaps the latter information could be ignored.

Choosing the add operator balances the cost in terms of memory search time with the possibility of short-circuiting the assembly process. Short-circuiting is an important feature of music creativity. For instance, in melody composition there is no chronological ordering between rhythm and pitches (Sloboda 1985). Sometimes, both occur together! This feature is often neglected by computational formalisms (Vincinaza & Prietula 1989).

### 5.3 - The Execution Module

The problem of planning in a dynamic world is that when the plan is being generated new events may occur and invalidate it. In music performance, it suffices that the musician plays to provoke changes in the context. Thus, monitoring context changes at the same time as replanning what is being executed is very difficult in real-time conditions.

In our model we consider that the reasoning mechanisms that underlie planning and replanning in music performance are not the same. The replanning that can be done while playing is related more to simple and

fast anatomic reactions than to complicated and refined reasoning. Consequently, beyond the role of controlling a MIDI synthesizer, the execution module has also to perform the changes in already generated plans. The idea is that particular context events trigger simple replanning such as "modify overall amplitude", "don't play these notes", "replace this note by another", etc. In short, since the composition module has finished its task, it is no longer concerned by changes to the plan it has generated. The context events occurring during a given plan generation will only be taken into account in the following plan generation.

At the current stage, the execution module has no replanning facilities. Notes are executed by a MIDI scheduler developed by Bill Walker (CERL Group - University of Illinois).

## 6 - Discussion

We have shown how an extension to classical problem solving could simulate some features of musical creativity. This extension attempts to incorporate both the experience musicians accumulate by practicing and the interference of the context in the musicians' ongoing reasoning. Although we do not use randomness in our model, there is no predetermined path to generate music. The musical result is constructed gradually by the interaction between the PACTs activated by the context and the Musical Memory's resources.

The notion of PACTs was first implemented (Pachet 1990) for the problem of generating live bass line and piano voicing. At this time, results were encouraging but, exploring exclusively a rule-based approach, various configurations of PACTs were hardly treated, if at all. This was due to the difficulty of expressing all musical choices in terms of rules. Our work has concentrated on improving the formalization of PACTs within a problem solving perspective. We have also introduced the notion of Musical Memory and seen how it can be coupled with PACTs. Today, Pachet's system is being reconsidered and re-implemented to take into account both the Musical Memory and a wider repertoire of PACTs.

In our model we have bypassed perceptual modeling. This is a tactical decision with respect to the complexity of modeling creativity in music. However, this modeling is essential for two reasons: to provide a machine with full creative behavior in music and, if coupled with machine learning and knowledge acquisition techniques, to help us in acquiring PACTs.

## Acknowledgments

We would like to thank François Pachet, Jean-Daniel Zucker and Vincent Corruble who, as both musicians and computer scientists, have given us continuous encouragement and technical support. This work has been partly supported by a grant from the Brazilian Education Ministry - CAPES/MEC.

## References

- AAAI '93 Workshop on Artificial Intelligence & Creativity 1993. Melon Park, AAAI Press.
- Ambros-Ingerson, J. & Steel, S. 1988. Integrating Planning, Execution and Monitoring, In Proceedings of the Sixth National Conference on Artificial Intelligence, 83-88, AAAI Press.
- Ames, C. & Domino, M. 1992. Cybernetic Composer: an overview, In M. Balaban, Ebicioğlu K. & Laske, O. eds., *Understanding Music with AI: Perspectives on Music Cognition*, The AAAI Press, California.
- Baker, M. 1980. Miles Davis Trumpet, Giants of Jazz Series, Studio 224 Ed., Lebanon.
- Baudoin, P. 1990 *Jazz: mode d'emploi*, Vol. I and II. Editions Outre Mesure, Paris.
- Johnson-Laird, P. 1992. *The Computer and the Mind*, Fontana, London.
- Laird, J., Newell, A. & Rosembloom, P. 1987. SOAR: An Architecture of General Intelligence, *Artificial Intelligence* 33, 1-64.
- Newell, A. & Simon, H. 1972. Human Problem-Solving, Englewood Cliffs. Prentice Hall, NJ.
- Nilsson, N. 1971. *Problem-Solving Methods in Artificial Intelligence*, McGraw-Hill Book Co., New York.
- Pachet, F. 1990. Representing Knowledge Used by Jazz Musicians, In the Proceedings of the International Computer Music Conference, 285-288, Montreal.
- Ramalho, G & Ganascia, J.-G. 1994. The Role of Musical Memory in Creativity and Learning: a Study of Jazz Performance, In M. Smith, Smaill A. & Wiggins G. eds., *Music Education: an Artificial Intelligence Perspective*, Springer-Verlag, London.
- Ramalho, G. & Pachet, F. 1994. What is Needed to Bridge the Gap Between Real Book and Real Jazz Performance?, in the Proceedings of the Fourth International Conference on Music Perception and Cognition, Liège.
- Rowe, J. & Partridge, D. 1993. Creativity: a survey of AI approaches, *Artificial Intelligence Review* 7, 43-70, Kluwer Academic Pub.
- Slade, S. 1991. Case-Based Reasoning: a Research Paradigm, *AI Magazine*, Spring, 42-55.
- Sloboda, J., 1985. *The Musical Mind: The Cognitive Psychology of Music*, Oxford University Press, New York.
- Steedman, M. 1984. A Generative Grammar for Jazz Chord Sequences, *Music Perception*, Vol. 1, No. 2, University of California Press.
- Vincinanza, S. & Prietula, M. 1989. A Computational Model of Musical Creativity, In Proceedings of the Second Workshop on Artificial Intelligence and Music, 21-25, IJCAI, Detroit.