Vertegal, R. and **Bonis**, E. (1994), ISEE: An Intuitive Sound Editing Environment, in *Computer Music Journal*, Vol. 18, Nr. 2, The MIT Press.

von Bismark, G. (1971), Timbre of Steady Sounds: Scaling of Sharpness, in *Proceedings of the 7*th *Internacional Congress on Acoustics in Budapest*, Vol. 3, pp. 637-640.

von Bismark, G. (1974a), Timbre of Steady Sounds: A Factorial Investigation of its Verbal Attributes, in *Acustica*, Vol. 30, pp. 146-158.

von Bismark, G. (1974b), Sharpness as an Attribute of the Timbre of Steady Sounds, in *Acustica*, Vol. 30, pp. 159-172.

Winston, P. (1984), *Artificial Intelligence*, (2nd ed.), Addison-Wesley.

Woodhouse, J. (1992), Physical Modeling of Bowed Strings, in *Computer Music Journal*, Vol. 16, Nr. 4, pp. 43-56, The MIT Press.

Xenakis, I. (1963), Musiques Formelles, in *La Revue Musicalle*, double issue Nrs. 253-254, Editions Richard-Masse.

Xenakis, I. (1971), *Formalized Music: Thought and Mathematics in Music Composition*, Indiana University Press.

Xenakis, I. (1992), *Formalized Music: Thought and Mathematics in Music*, Pendragon Press, revised edition.

Representing Musicians' Actions for Simulating Improvisation in Jazz

Geber Ramalho

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-mail: ramalho@laforia.ibp.fr

Abstract

This paper considers the problem of simulating 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 exploits the principles of Case-Based Reasoning (Schank & Riesbeck 1989). To produce live music using this *Musical Memory* we propose a problem solving method based on the notion of PACTs *(Potential ACTions)* (Ramalho & Ganascia 1994b). These PACTs are a generic framework for representing the musical actions that are activated according to the context and then combined in order to produce notes.

1 - Introduction

This paper considers the problem of simulating the behavior of a bass player in the context of Jazz live performance. 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 the notion of PACTs, the basic element of our model. In Section 4, we give a general description of our model and show 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

The tasks of improvisation and accompaniment consist in playing notes (melodies and/or chords) according to guidelines laid down in a given chord grid (sequence of chords underlying the song). 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 large gap referred to above (Ramalho & Pachet 1994). 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). Regardless of its musical results, the random-based approach

cannot provide an accurate understanding of musical knowledge, since no explicit semantics is associated to randomness. On the other hand, the deterministic framework of the logic-based approach lacks of flexibility for modeling musical creativity. 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

Our first claim is that Jazz musicians' activities are supported by two main knowledge structures: memories and rules. Jazz musicians use rules they have learned in schools and through Jazz methods (Baudoin 1990). However, these rules do not embody all knowledge. In fact, despite the availability of some rules for manipulating abstract concepts such as tension, style, swing, contour, density, contrast, etc., there is no logical rule chaining that can directly and uniquely instantiate these concepts 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).

To put it in a nutshell, musicians integrate rules and memories into their actions dynamically (Ramalho & Ganascia 1994a). Sometimes, the notes can be determined from their most abstract concepts by means of rules 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. This memory search is both flexible and controlled because of the mechanism of partial matching between the memory contents and requirements stated the available rules and concepts. 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 - PACTs: the Basic Notion of our Model

3.1 - Introduction

Pachet (Pachet 1990) has proposed the notion of PACTs (Potential ACTions) as a generic framework for representing the potential actions 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. It is this homogeneous representation of both notes and their related abstract concepts PACTs that allows the integration of analytic (rule-based) and analogical (case-based) reasoning.

More precisely, PACTs are frame-like structures whose *main attributes* are: start-beat, end-beat, dimensions, abstract-level, type and instrument-dependency. PACTs are activated at a precise moment in time and are of limited duration which can correspond to 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. 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 *declarative* (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.).

3.2 - PACTs as basis of the problem solving method
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

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 (Newell & Simon 1972; Nilsson 1971). Taking an initial state of a problem space as a time segment (e.g. bars) with no notes, a musical problem could consist in filling this time segment with notes which satisfy some criteria. This intuitive formulation of what a musical problem is (Vincinanza & Prietula 1989) has been criticized by many researchers because these criteria are not determined *a priori* (Johnson-Laird 1992). 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 "vague criteria" and go towards a precise specification of these criteria. In other words, solving a musical problem consists in assembling (combining) a set of PACTs that have been activated by the performance context. The goal is fixed and clearly defined (i.e. the goal is to play!).

3.3 - PACTs as the contents of the Musical Memory

There is no guarantee that a set of PACTs contains the necessary information so as to produce a playable PACT. To solve this problem we have introduced the notion of Musical Memory which explores the principles of case-based reasoning (Schank & Riesbeck 1989). This Musical Memory is a long term memory that accumulates the musical material (cases) the musicians have listened to. These cases are represented using PACTs' framework and thus can be retrieved and modified during the problem solving to provide missing information.

The 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 94a). These indexes are in fact the same attributes used to describe activated PACTs. For instance, pitches are described in terms of contour, tessitura, set and style as discussed in last section.

Low-level PACTs are PACTs whose attributes are all specialized, i.e. have defined values. Describing a Musical Memory case in terms of PACTs correspond to start from note-level attribute to fill in the more abstract ones. Whereas, in the process of assembling PACTs we start from abstract descriptions to combine them into note-level ones. When this latter is not possible, a match between the already specialized attributes and the PACTs in the Musical memory is performed.
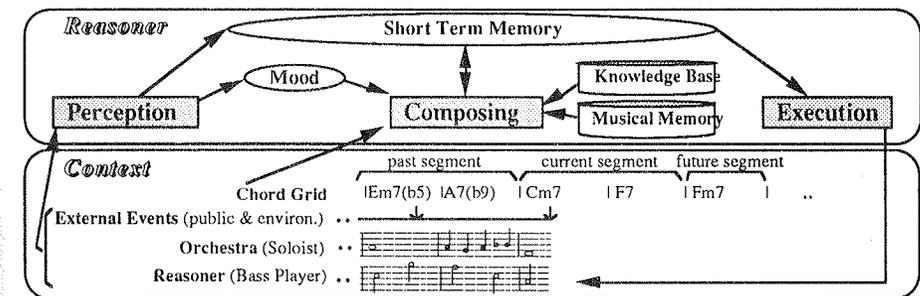

Figure 1 - Overall Description of the Model

4 - General Description of our Model

4.1 - The Reasoner and the Composing Module

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)

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

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.

In next sections we give further details of the composition module. The discussion of perception and the execution modules is not in the scope of this paper (see Ramalho & Ganascia 1994b).

4.2 - 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.

4.3 - 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 & Rosembloom, 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 - 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 using a Smalltalk platform to take into account both the Musical Memory and a wider repertoire of PACTs.

References

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, Ebicioglu 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 Mésure, 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. (1994a). 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 & Ganascia, J.-G. (1994b). Simulating Creativity in Jazz Performance, à paraître dans Proceedings of the Twelfth National Conference in Artificial Intelligence - AAAI '94, Seattle, AAAI Press.

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.

Schank, R. & Riesbeck, C. (1989). Inside Case-base Reasoning, lawrence Erlaum Assoc. Pub., New Jersey.

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.

Desempenho, Interface com o Usuário e Projeto de Instrumentos