Applied Intelligence 14, 115–129, 2001

An Interactive Case-Based Reasoning Approach for Generating Expressive Music

JOSEP LLUÍS ARCOS AND RAMON LÓPEZ DE MÁNTARAS
Artificial Intelligence Research Institute, Spanish Council for Scientific Research, Campus UAB, 08193 Bellaterra, Catalonia, Spain
arcos@iiia.csic.es
mantaras@iiia.csic.es

Abstract. In this paper we present an extension of an existing system, called *SaxEx*, capable of generating expressive musical performances based on Case-Based Reasoning (CBR) techniques. The previous version of *SaxEx* used pre-fixed criteria within the different CBR steps and, therefore, there was no room for user interaction. This paper discusses the necessity of user interaction during the CBR process and how this decision enhances the capabilities and the usability of the system. The set of evaluation experiments conducted show the advantages of *SaxEx*'s new interactive functionality, particularly for future educational applications of the system.

Keywords: Case-Based Reasoning, user interaction, musical expression

1. Introduction

The work described in this paper addresses the generation of expressive music, endowing the resulting piece with the expressivity that characterizes human performances. Following musical rules, independently of how sophisticated and complete they are, is not enough to achieve this expressivity, and indeed music generated in this way usually sounds monotonous and mechanical. The main problem here is to grasp the performer's "personal touch", the knowledge brought about when performing a score and that is absent from it. This knowledge concerns not only "technical" features (use of musical resources) but also the affective aspects implicit in music. A large part of this knowledge is tacit and therefore very difficult to generalize and verbalize, although it is not inaccessible. Humans acquire it through a long process of observation, imitation, and experimentation [1].

For this reason, AI approaches based on declarative knowledge representations have serious limitations. An alternative approach, much closer to the observation-imitation-experimentation process observed in humans, is that of directly using the knowledge implicit in examples from recordings of human performances.

To achieve this we developed *SaxEx* [2], a case-based reasoning (CBR) system for generating expressive performances of melodies based on examples of human performances (for the moment *SaxEx* is limited to jazz ballads). CBR [3] is appropriate for problems where (a) many examples of solved problems can be obtained—like in our case where multiple examples can be easily obtained from recordings of human performances; and (b) a large part of the knowledge involved in the solution of problems is tacit, difficult to verbalize and generalize.

Previous versions of *SaxEx* used pre-fixed criteria within the different CBR steps and, therefore, there was no room for user interaction. In this version we have improved the CBR component by allowing the user to interact with and to influence the CBR process.

User interaction is necessary because on the one hand, generating expressive performances is a creative process and as such it can certainly be enhanced by human intervention, especially in its aesthetic evaluation [4]. On the other hand, since we focus on its use as an educational tool for music students, it was necessary to provide the tool with flexible experimentation capabilities.

This paper is organized as follows: The next section describes the elements from which the system has been built; Section 3 describes how the system works; Section 4 focuses on the interaction capabilities of the system; Section 5 presents the results obtained during the evaluation of the system; Section 6 comments on some related work; and, finally, in Section 7 we give some conclusions and point to future work.

2. SaxEx Elements

In this section, we briefly present some of the elements underlying *SaxEx* that are necessary to understand the system (see Fig. 1).

2.1. SMS

Sound analysis and synthesis techniques based on spectrum models like Spectral Modeling and Synthesis (SMS) are useful for the extraction of high level parameters from real sound files, their transformation, and the synthesis of a modified version of these sound files. *SaxEx* uses SMS in order to extract basic information related to several expressive parameters such as

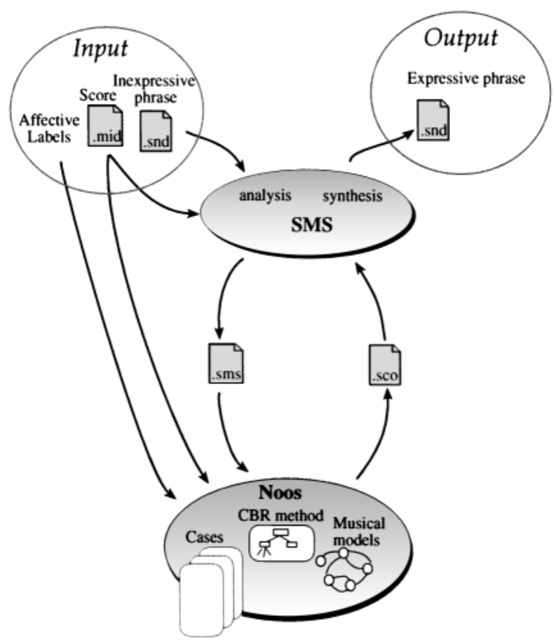

Figure 1. General view of *SaxEx* blocks.

dynamics, rubato, vibrato, and articulation. The SMS synthesis procedure allows the generation of expressive reinterpretations by appropriately transforming an inexpressive sound file.

The SMS approach to spectral analysis is based on decomposing a sound into sinusoids plus a spectral residual. From the sinusoidal plus the residual representation we can extract high level attributes such as attack and release times, formant structure, vibrato, and average pitch and amplitude, when the sound is a note or a monophonic phrase of an instrument. These attributes can be modified and added back to the spectral representation without loss of sound quality.

This sound analysis and synthesis system is ideal as a preprocessor, giving to *SaxEx* high level musical parameters, and as a post-processor, adding the transformations specified by the case-based reasoning system to the inexpressive original sound.

2.2. Noos

SaxEx is implemented in *Noos* [5, 6], a reflective object-centered representation language designed to support knowledge modeling of problem solving and learning. Modeling a problem in Noos requires the specification of three different types of knowledge: domain knowledge, problem solving knowledge, and metalevel knowledge.

Domain knowledge specifies a set of concepts, a set of relations among concepts, and problem data that are relevant for an application. Concepts and relations define the domain ontology of an application. For instance, the domain ontology of *SaxEx* is composed of concepts such as notes, chords, analysis structures, and expressive parameters. Problem data, described using the domain ontology, define specific situations (specific problems) that have to be solved. For instance, specific inexpressive musical phrases to be transformed into expressive ones.

Problem solving knowledge specifies the set of tasks to be solved in an application. For instance, the main task of *SaxEx* is to infer a sequence of expressive transformations for a given musical phrase. Methods model different ways of solving tasks. Methods can be elementary or can be decomposed into subtasks. These new (sub)tasks may be achieved by other methods. A method defines an execution order over subtasks and a specific combination of the results of the subtasks in order to solve the task it performs. For a given task, there can be multiple alternative methods that may solve the task in different situations. This recursive decomposition of a task into subtasks by means of a method is called task/method decomposition.

The metalevel of Noos incorporates, among other types of (meta-)knowledge, *Perspectives*, used in the retrieval task, and *Preferences*, used by *SaxEx* to rank cases.

2.2.1. Perspectives. These [7] constitute a mechanism for describing declarative biases for case retrieval for structured and complex case representations. They provide a flexible and dynamic retrieval mechanism and are used by *SaxEx* to make decisions about the relevant aspects of a problem. *SaxEx* incorporates two types of declarative biases in the perspectives. On the one hand, metalevel knowledge to assess similarities among scores using the analysis structures built upon musical models (described below in Section 2.3). On the other hand, (metalevel) knowledge to detect affective intention in performances and to assess similarities among them.

2.2.2. Preferences. Model decision making criteria about sets of alternatives present in domain knowledge and problem solving knowledge. In the *SaxEx* context, preferences are used as a symbolic representation of relevance (or "similitude") in comparing a given current problem with problems previously solved by the system. For instance, preference knowledge can be used to model criteria for ranking some precedent cases over other precedent cases for a task in a specific situation. Preferences are modeled by partially ordered sets (also called *posets*) and are built by means of *preference methods*. There are two kinds of preference methods: preference construction methods and preference combination methods.

A preference construction method takes a set of source elements and an ordering criterion and builds a partially ordered set. *Noos* provides several built-in preference constructors based on numerical and non-numerical criteria. Examples of preference construction methods are `increasing-preference` and `decreasing-preference` that take a set of elements with a common numeric feature and build a preference where the preferred elements are those with a greater value or lesser value, respectively, in the specified feature. Some examples are shown in Section 4.2.

There are several ways to combine different preference criteria—or, in other words, build new preferences from existing preferences. The *Noos* operations dealing with preference combination are methods that create new partially ordered sets from (a combination of) partially ordered sets—created either by preference construction methods or by other preference combination methods. Examples of preference combinations are operations such as `inversion`, `preference-union`, `preference-intersection`, and `hierarchical-preference-union`.

For instance, `preference-union` takes two preference criteria and constructs a new preference performing a union of the elements of the sets and a transitive closure of the union of order relations. As we will show in Section 4.2, `preference-union` is used to combine precedents obtained from equally preferred perspectives such as melodic direction and note duration. Another example of a preference combination used by *SaxEx* is `hierarchical-preference-union` that, given a more preferred poset called `higher-poset` and a less preferred poset called `lower-poset`, constructs a preference order preserving the order fixed in `higher-poset` and adding from `lower-poset` the order relations that are not in conflict with `higher-poset`. `hierarchical-preference-union` is used by *SaxEx* to combine precedents obtained from perspectives with different preference.

2.2.3. Episodic Memory. This is the (accessible and retrievable) collection of problems that the system has solved. Once a problem is solved, *Noos* provides a collection of special methods to dynamically indicate which problems must be stored and which problems can be forgotten. Using these methods, *Noos* applications can automatically incorporate (store and index) some problems into the episodic memory.

Noos provides a set of basic retrieval methods that can retrieve previous relevant episodes from the episodic memory using relevance criteria. Relevance criteria are determined by specific domain knowledge about the importance of different features or by requirements of problem solving methods. Usually, the notion of similitude in case-based reasoning introduces a way to assess the relevance of precedent cases in solving a new case. Similarity measures estimate a relevance order between precedent cases. Our approach is to work directly over relevance orders.

Retrieval methods are based on the notion of feature terms as partial descriptions and the notion of subsumption among feature terms [8]. The intuitive meaning of subsumption is that a term t_1 subsumes another term t_2 ($t_1 \sqsubseteq t_2$) when t_1 is more general than t_2. Notice that we treat subsumption ordering $\sqsubseteq$ as an informational ordering i.e., ($t_1 \sqsubseteq t_2$ means that t_1 has less or equal information content than t_2). Our approach is that a knowledge modeling analysis can determine the relevant aspects of problems; then, partial descriptions of the current problem can be built embodying the aspects considered as relevant. These partial descriptions are used as retrieval patterns for searching similar cases in the episodic memory using subsumption. Thus, retrieval methods can be viewed as methods that search into the episodic memory for the set of feature terms subsumed by a feature term, a pattern, embodying the relevant aspects of a problem data. Retrieval methods are the basic building block for integrating learning, and specifically CBR, into Noos. Retrieval methods, as we will show in Section 3.2, are used by *SaxEx* in the *search* subtask.

2.3. Background Musical Knowledge

SaxEx incorporates two general theories of musical perception and musical understanding: Narmour's implication/realization (IR) model [9] and Lerdahl and Jackendoff's generative theory of tonal music (GTTM) [10]. Moreover, *SaxEx* incorporates specific knowledge about Jazz theory. These three musical models constitute the background musical knowledge of the system.

2.3.1. Narmour's Implication/Realization Model. This proposes a theory of cognition of melodies based on eight basic structures. These structures characterize patterns of melodic implications that constitute the basic units of the listener perception. Other parameters such as metric, duration, and rhythmic patterns emphasize or inhibit the perception of these melodic implications. The use of the IR model provides a musical analysis based on the structure of the melodic surface.

Examples of IR basic structures are the `P` process (a melodic pattern describing a sequence of at least three notes with similar intervals and the same ascending or descending registral direction) and the `ID` process (a sequence of at least three notes with the same intervals and different registral directions), among others.

2.3.2. Lerdahl and Jackendoff's Generative Theory of Tonal Music. GTTM, on the other hand, offers a complementary approach to understanding melodies based on a hierarchical structure of musical cognition. GTTM proposes four types of hierarchical structures associated with a piece: the *grouping structure*, the *metrical structure*, the *time-span reduction structure*, and the *prolongational reduction structure*.

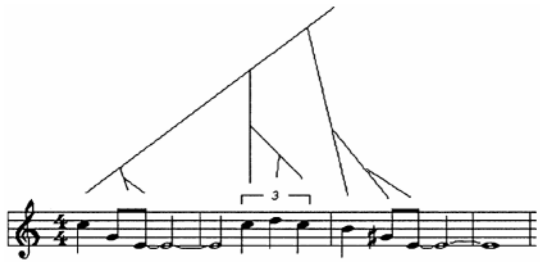

Figure 2. Example of a time-span tree for the beginning of the 'All of me' ballad.

The *grouping structure* describes the segmentation units that listeners can establish when hearing a musical surface: motives, phrases, and sections. The *metrical structure* describes the rhythm hierarchy of the piece. The *time-span reduction structure* is a hierarchical structure describing the relative structural importance of notes within the audible rhythmic units of a phrase (see Fig. 2). The *prolongational reduction structure* is a hierarchical structure describing tension-relaxation relationships among groups of notes.

The grouping structure can help to determine the phrase level. The metrical structure is represented in *SaxEx* associating a metrical-strength to each note. The time-span reduction structure and the prolongational-reduction structure are tree structures that are directly represented in *Noos* because of the tree-data representation capabilities of the language.

The goal of using both IR and GTTM models is to take advantage of combining the IR analysis of melodic surface with the GTTM structural analysis of the melody. These are two complementary views of melodies that influence the execution of a performance.

2.3.3. Jazz Theory. This is introduced in *SaxEx* for the specific treatment of harmony in jazz. In jazz the notion of tonality is secondary and other aspects such as chord progressions, the tonal functionality of chords, or the use of dominants are more important. Since we are using *SaxEx* for generating expressive performances of jazz ballads, jazz theory is useful to determine harmonic stability of notes and the role of the notes with respect to the underlying harmony.

3. The SaxEx System

An input for *SaxEx* (see Fig. 1) is a musical phrase described by means of its musical score (a MIDI file), a sound, and specific qualitative values along three affective dimensions (tender-aggressive, sad-joyful, calm-restless) expressing the user preferences regarding the desired expressive output performance [11]. Affective information can be partially specified, that is the user does not have to provide values for every dimension. The score contains the melodic and the harmonic information of the musical phrase. The sound contains the recording of an inexpressive interpretation of the musical phrase played by a musician. Values for affective dimensions will guide the search in the memory of cases.

The output of the system is a set of new sound files, obtained via transformations of the original sound, where each contains a different expressive performance of the same phrase according to the affective labels given as input.

Solving a problem in *SaxEx* involves three phases: the analysis phase, the reasoning phase, and the synthesis phase. The Analysis and synthesis phases are implemented using SMS sound analysis and synthesis techniques. The reasoning phase is performed using CBR techniques, implemented in Noos, and is the main focus of this paper.

The development of *SaxEx* involved the elaboration of two main models: the domain model and the problem-solving model. The domain model contains the concepts and structures relevant for representing musical knowledge. The problem-solving model consists mainly of a CBR method for inferring a sequence of expressive transformations for a given musical phrase.

3.1. Modeling Musical Knowledge

Problems solved by *SaxEx*, and stored in its memory, are represented as complex structured cases embodying three different kinds of musical knowledge (see Fig. 3): (1) concepts related to the score of the phrase such as notes and chords, (2) concepts related to background musical theories such as implication/realization structures and GTTM's time-span reduction nodes, and (3) concepts related to the performance of musical phrases.

A score is represented by a melody, embodying a sequence of notes, and a harmony, embodying a sequence of chords. Each note holds in turn a set of features such as its pitch (C5, G4, etc), its position with respect to the beginning of the phrase, its duration, a reference to its underlying harmony, and a reference to the next note of the phrase. Chords hold also a set of features such as name (Cmaj7, E7, etc), position, duration, and a reference to the next chord.

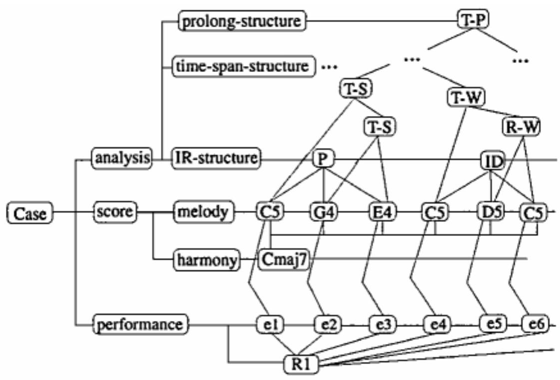

Figure 3. Overall structure of the beginning of an 'All of me' case.

The musical analysis representation embodies structures of the phrase automatically inferred by *SaxEx* from the score using IR and GTTM background musical knowledge. The analysis structure of a melody is represented by a process-structure (embodying a sequence of IR basic structures), a time-span-reduction structure (embodying a tree describing metrical relations), and a prolongational-reduction structure (embodying a tree describing tensing and relaxing relations among notes). Moreover, a note holds the metrical-strength feature, inferred using GTTM theory, expressing the note's relative metrical importance into the phrase. Section 3.3 describes in more detail these structures.

The information about the expressive performances contained in the examples of the case memory is represented by a sequence of *affective regions* and a sequence of *events*, one for each note (extracted using the SMS sound analysis capabilities), as explained below.

Affective regions group (sub)-sequences of notes with common affective expressivity. Specifically, an affective region holds knowledge describing the following affective dimensions: *tender-aggressive*, *sad-joyful*, and *calm-restless*. These affective dimensions are described using five ordered qualitative values expressed by linguistic labels as follows: the middle label represents no predominance (for instance, neither tender nor aggressive), lower and upper labels represent, respectively predominance in one direction (for example, absolutely calm is described with the lowest label). For instance, a jazz ballad can start very tender and calm and continue very tender but more restless. Such different nuances are represented in *SaxEx* by means of different affective regions.

There is an *event* for each note within the phrase embodying information about expressive parameters applied to that note. Specifically, an event holds information about dynamics, rubato, vibrato, articulation, and attack. These expressive parameters are described using qualitative labels as follows:

- Changes in dynamics are described relative to the average loudness of the phrase by means of a set of five ordered labels. The middle label represents average loudness and lower and upper labels represent respectively increasing or decreasing degrees of loudness.
- Changes in rubato are described relative to the average tempo also by means of a set of five ordered labels. Analogously to dynamics, qualitative labels about rubato cover the range from a strong accelerando to a strong ritardando.
- The vibrato level is described using two parameters: frequency and amplitude. Both parameters are described using five qualitative labels from no-vibrato to highest-vibrato.
- The articulation between notes is described using again a set of five ordered labels covering the range from legato to staccato.
- Finally, *SaxEx* considers two possibilities regarding note attack: (1) reaching the pitch of a note starting from a lower pitch, and (2) increasing the noise component of the sound. These two possibilities were chosen because they are characteristic of saxophone playing but additional possibilities could be introduced without altering the system.

3.2. The SaxEx CBR Task

The task of *SaxEx* is to infer a set of expressive transformations to be applied to every note of an inexpressive phrase given as input. To achieve this, *SaxEx* uses a CBR problem solver, a case memory of expressive performances, and background musical knowledge. Transformations concern the dynamics, rubato, vibrato, articulation, and attack of each note in the inexpressive phrase. The cases stored in the episodic memory of *SaxEx* contain knowledge about the expressive transformations performed by

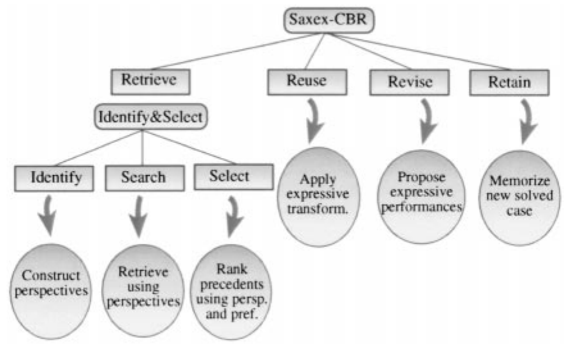

Figure 4. Task decomposition of the *SaxEx* CBR method.

a human player given specific labels for affective dimensions.

For each note in the phrase, the following subtask decomposition (Fig. 4) is performed by the CBR problem solving method implemented in Noos:

- *Retrieve*: The goal of the retrieve task is to choose, from the memory of cases (pieces played expressively), the set of precedent notes—the cases—most similar for every note of the problem phrase. Specifically, the following subtask decomposition is applied to each note of the problem phrase:
  - *Identify*: its goal is to build retrieval perspectives using the affective values specified by the user and the musical background knowledge integrated in the system (retrieval perspectives are described in Subsection 3.3). These perspectives guide the retrieval process by focusing it on the most relevant aspects of the current problem, and will be used either in the *search* or in the *select* subtasks.
  - *Search*: its goal is to search cases in the case memory using Noos retrieval methods and some previously constructed perspective(s).
  - *Select*: its goal is to rank the retrieved cases using Noos preference methods. The collection of *SaxEx* default preference methods use criteria such as similarity in duration of notes, harmonic stability, or melodic directions.
- *Reuse*: its goal is to choose, from the set of most similar notes previously retrieved, a set of expressive transformations to be applied to the current note. The default strategy of *SaxEx* is the following: the first criterion used is to adapt the transformations of the most similar note. When several notes are considered

equally similar, the transformations are selected according to the majority rule. Finally, in case of a tie, one of them is selected randomly (reuse criteria are described in Subsection 3.4). When the retrieval task is not able to retrieve similar precedent cases for a given note, no expressive transformations are applied to that note and the situation is notified in the revision task. Nevertheless, using the current *SaxEx* case base, the retrieval perspectives allways retrieved at least one precedent in the experiments performed.

- *Revise*: its goal is to present to the user a set of alternative expressive performances for the problem phrase. As we will describe in the next section, users can tune the expressive transformations applied to each note and can indicate which performances they prefer.
- *Retain*: the incorporation of the new solved problem to the memory of cases is performed automatically in *Noos* from the selection performed by the user in the *revise* task. These solved problems will be available for the reasoning process when solving future problems. Only positive feedback is given. That is, only those examples that the user judges as good expressive interpretations are actually retained.

In previous versions of *SaxEx* the CBR task was fixed. That is, the collection of retrieval perspectives, their combination, the collection of reuse criteria, and the storage of solved cases were pre-designed and the user didn't participate in the reasoning process. Moreover, the *retain* subtask was not present because it is mainly a subtask that requires an interaction with the user.

Now, in the current version of *SaxEx* we have improved the CBR method by incorporating the user in the reasoning process. This new capability allows users to influence the solutions proposed by *SaxEx* in order to satisfy their interests or personal style. The user can interact with *SaxEx* in the four main CBR subtasks. This new functionality requires that the use and combination of the two basic mechanisms—perspectives and preferences—in the Retrieve and Reuse subtasks must be parameterizable and dynamically modifiable. That is, pre-fixed criteria cannot be implemented in this version of *SaxEx*.

Below we will present the collection of retrieval and reuse criteria provided by *SaxEx*. Then, in the next section we will present how the user can interact with the system in order to tailor the behavior of *SaxEx* by either activating/deactivating criteria or combining them in a specific way.

3.3. Retrieval Perspectives

Retrieval perspectives are built by the *identify* subtask and can be used either by the *search* or the *select* subtask. Perspectives used by the *search* subtask will act as filters. Perspectives used by the *select* subtask will act only as a preference. Retrieval perspectives are built based on user requirements and background musical knowledge. Retrieval perspectives provide partial information about the relevance of a given musical aspect. After these perspectives are established, they have to be combined in a specific way according to the importance (preference) that they have.

Retrieval perspectives are of two different types: based on the affective intention that the user wants to obtain in the output expressive sound or based on musical knowledge.

(1) *Affective labels* are used to determine the following declarative bias: we are interested in notes with affective labels similar to the affective labels required in the current problem by the user.

As an example, let us assume that we declare we are interested in forcing *SaxEx* to generate a calm and very tender performance of the problem phrase. Based on this bias, *SaxEx* will build a perspective specifying as relevant to the current problem the notes from cases that belong first to "calm and very tender" affective regions (most preferred), or "calm and tender" affective regions, or "very calm and very tender" affective regions (both less preferred).

When this perspective is used in the *Search* subtask, *SaxEx* will search in the memory of cases for notes that satisfy this criterion. When this perspective is used in the *Select* subtask, *SaxEx* will rank the previously retrieved cases using this criterion.

(2) *Musical knowledge* gives three sets of declarative retrieval biases: first, biases based on Narmour's implication/realization model; second, biases based on Lerdahl and Jackendoff's generative theory; and third, biases based on Jazz theory and general music knowledge.

Regarding Narmour's implication/realization model, *SaxEx* incorporates the following three perspectives:

- The "*role in IR structure*" criterion determines as relevant the role that a given note plays in an implication/realization structure. That is, the kind of IR structure it belongs to (e.g., the `P` process described in Section 2.3.1) and its position (`first-note`, `inner-note`, or `last-note`). For instance, this retrieval perspective can specify biases such as "look for notes that are the `first-note` of a `P` process".

- The "*Melodic Direction*" criterion determines as relevant the kind of melodic direction in an implication/realization structure: `ascendant`, `descendant`, or `duplication`. This criterion is used for adding a preference among notes with the same IR role and different melodic direction.
- The "*Durational Cumulation*" criterion determines as relevant the presence—in an IR structure—of a note in the last position with a duration significally higher than the others. This characteristic emphasizes the end of an IR structure. This criterion is used—as the previous—for adding a preference among notes with the same IR role and different melodic direction.

Regarding Lerdahl and Jackendoff's GTTM theory, *SaxEx* incorporates the following three perspectives:

- The "*Metrical Strength*" criterion determines as relevant the importance of a note with respect to the metrical structure of the piece. The metrical structure assigns a weight to each note according to the beat in which it is played. That is, the metrical weight of notes played in strong beats are higher than the metrical weight of notes played in weak beats. For instance, the metrical strength bias determines as similar the notes played at the beginning of subphrases since the metrical weight is the same.
- The "*role in the Time-Span Reduction Tree*" criterion determines as relevant the structural importance of a given note according to the role that the note plays in the analysis Time-Span Reduction Tree.

  Time-Span Reduction Trees are built bottom-up and hold two components: a segmentation into hierarchically organized rhythmic units and a binary tree that represents the relative structural importance of the notes within those units. There are two kinds of nodes in the tree: left-elaboration nodes and right-elaboration nodes.

  Since the Time-Span Reduction Tree is a tree with high depth, we are only taking into account the two last levels. That is, given a note this perspective focuses on the kind of leaf the note belongs (left or right leaf) and on the kind of node the leaf belongs (left-elaboration or right-elaboration node).
  For instance, in the 'All of me' ballad (see Fig. 2) the first quarter note of the second bar (`C`) belongs to a left leaf in a right-elaboration node because the following two notes (`D` and `C`) elaborate the first note. In turn, these two notes belong to a left-elaboration (sub)node because the second note (`D`) elaborates the third (`C`).
- The "*role in the Prolongational Reduction Tree*" criterion determines as relevant the structural importance of a given note according to the role that the note plays in the Prolongational Reduction Tree. Prolongational Reduction Trees are binary trees built top-down and represent the hierarchical patterns of tension and relaxation among groups of notes. There are two basic kinds of nodes in the tree (tensing nodes and relaxing nodes) with three modes of branch chaining: *strong prolongation* in which events repeat maintaining sonority (e.g., notes of the same chord); *weak prolongation* in which events repeat in an altered form (e.g., from `I` chord to `I6` chord); and *jump* in which two completely different events are connected (e.g., from `I` chord to `V` chord).

  As in the previous perspective we are only taking into account the two last levels of the tree. That is, given a note this perspective focuses on the kind of leaf the note belongs (left or right leaf), on the kind of node the leaf belongs (tensing or relaxing node), and the kind of connection of the node (strong, weak, or jump).

Finally, regarding perspectives based on jazz theory and general music knowledge, *SaxEx* incorporates the following two:

- The "*Harmonic Stability*" criterion determines as relevant the role of a given note according to the underlying harmony. Since *SaxEx* is focused on generating expressive music in the context of jazz ballads, the general harmonic theory has been specialized taking harmonic concepts from jazz theory. The Harmonic Stability criterion takes into account in the following two aspects: the position of the note within its underlying chord (e.g., first, third, seventh, ...); and the role of the note in the chord progression it belongs.
- The "*Note Duration*" criterion determines as relevant the duration of a note. That is, given a specific situation, the set of expressive transformations applied to a note will differ depending on whether the note has a long or a short duration.

3.4. *Reuse Criteria*

As we have described in Section 3.2, the *reuse* task takes the ordered set (possibly partially ordered) of note precedents selected by the *retrieve* task and decides the expressive transformations to be performed to each note. That is, for every note in the problem phrase, we have to determine a value for each of the five expressive parameters. For instance, average loudness, strong accelerando, low vibrato frequency, etc.

Given a note and an expressive parameter, the first decision is to choose how many precedent notes have to be considered for reuse. Since retrieval perspectives model the similarity between problem notes and precedent notes, the *reuse* task selects the most similar precedent notes. Given this set of precedents the easiest situation is when all precedents have performed the same transformation in an expressive parameter. In that case, there is no conflict about the transformation to be applied to the problem note.

But this ideal situation is not usual. Usually the transformation applied in each precedent is not the same. Then, *SaxEx* decides which transformation to apply using some of the following reuse criteria where the first four are mutually exclusive as well as the fifth and sixth:

- The "*Majority Rule*" criterion chooses the values that were applied in the majority of precedents.
- The "*Strict Majority Rule*" criterion chooses the values that were applied in at least half of the precedents.
- The "*Minority Rule*" criterion chooses the values that were applied in the minority of precedents.
- The "*Strict Minority Rule*" criterion chooses the values that were applied in at most one of the precedents.
- The "*Continuity*" criterion gives priority to precedent notes belonging to the same musical subphrase in the case base.
- The "*Non-Continuity*" criterion is the inverse of the previous one. That is, it gives priority to precedent notes not belonging to the same musical subphrase in the case base.
- The "*Random*" criterion chooses randomly one value among precedent values. This criterion is used as the last criterion when after applying previous criteria more than one alternative remains.

The default strategy of *SaxEx* uses first the majority rule and if necessary the random criterion. Nevertheless, since the generation of expressive performances is a creative process mainly influenced by the user's personal preferences, the other reuse criteria can be used to taylor the system according to those personal preferences. For instance, the "*Strict Minority Rule*" and the "*Non-Continuity*" criteria will force *SaxEx* to produce expressive performances containing less usual combinations of expressive effects. As we will show in Section 4.2, other the criteria can be selected and combined by users.

4. Interacting with SaxEx

In the previous section we have described the CBR process performed by *SaxEx* (Section 3.2) and the collection of basic criteria used by taking decisions (Sections 3.3 and 3.4). Now we will describe how a user interacts with the system and influences the CBR process. A typical interaction scenario between an apprentice user and *SaxEx* could be the following: The user launches *SaxEx* and an initial panel appears requesting several pieces of information (see Fig. 5). The user then chooses the musical phrase to be generated and some affective values, clicks the start button, and a second panel (see Fig. 7) shows the proposed solutions, allowing the user to listen to them. Probably some proposed solutions satisfy better than others the user's expectations and, therefore, the user wishes to understand or improve (according to her personal style) the solutions provided by the system. At that point, the user is actually ready to participate in the CBR reasoning process.

Users can interact with *SaxEx* in all the four main CBR tasks: by deciding which criteria to use in the case retrieval (see Fig. 6), by deciding how the solutions from precedents should be used in the current problem (see Fig. 6), revising the solutions provided by the system (see Fig. 7), and deciding which problems must be retained in the memory of cases—that is, which problems will influence the resolution of future problems (see Fig. 7).

The interaction with *SaxEx* is therefore organized in three panels: a panel for specifying the problem to be solved (the musical phrase and the affective values); a panel for manipulating the criteria used in the retrieval and the reuse tasks; and a panel that shows the proposed solutions, allows the user to revise them, and allows the user to select which one to retain. We now describe these interactions in more detail.

4.1. Specifying a New Problem

First of all, users have to select a musical problem phrase (see Fig. 5). Users can choose from a set of pre-existing ballads or can provide a new one. In the case of a new ballad, the user has to provide a MIDI file

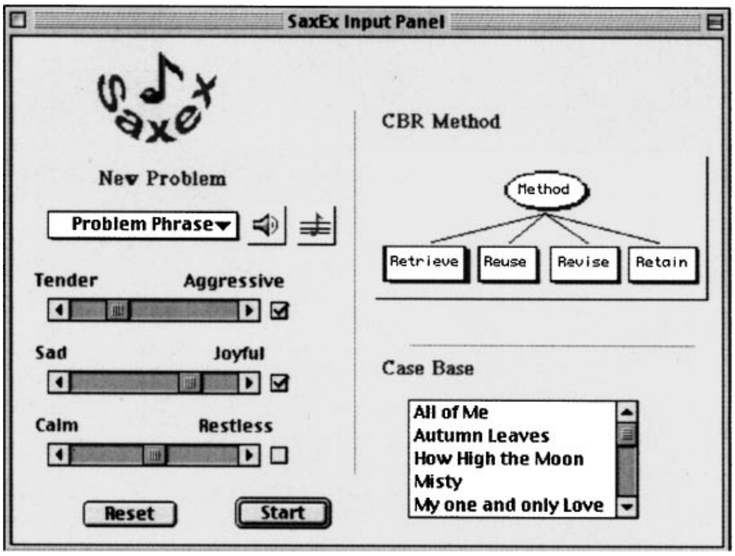

Figure 5. *SaxEx* panel for specifying a new problem.

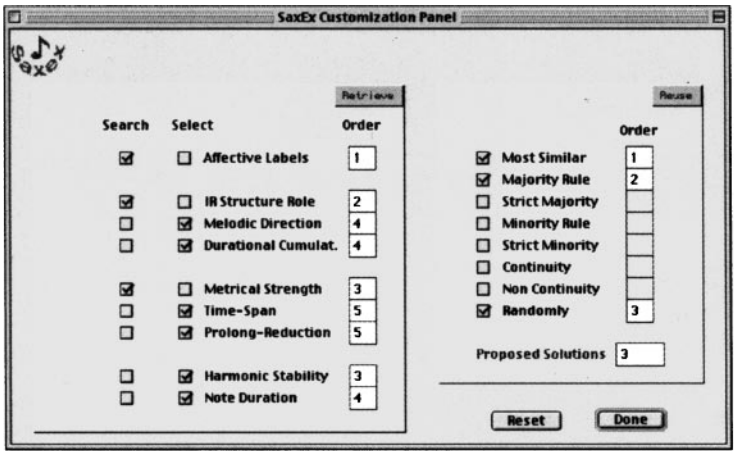

Figure 6. *SaxEx* panel for customizing the reuse and retain tasks.

containing the score (the melody and the harmony) and a sound file containing a recording of an inexpressive interpretation of the musical phrase played by a musician. After this selection, the musical phrase can be played and its score can be also displayed.

Next, some values along the three affective dimensions can be specified. Affective labels can be partially specified, i.e., the user does not have to provide labels for every dimension. In order to really specify a value (using the slider shown in Fig. 5) for an affective dimension, the dimension has to be activated (using the checkbox button).

Since we can choose any ballad from the existing *SaxEx* collection, we have to determine which of the remaining ballads will be used as cases.

Finally, the user can either click on the retrieval or reuse tasks to manipulate the default strategy of the system, or click on the start button and proceed to the Interactive Revision panel to customize the CBR cycle.

4.2. *Customizing Retrieve and Reuse*

The customization panel (Fig. 6) is divided into two subpanels: retrieval and reuse. The goal in both

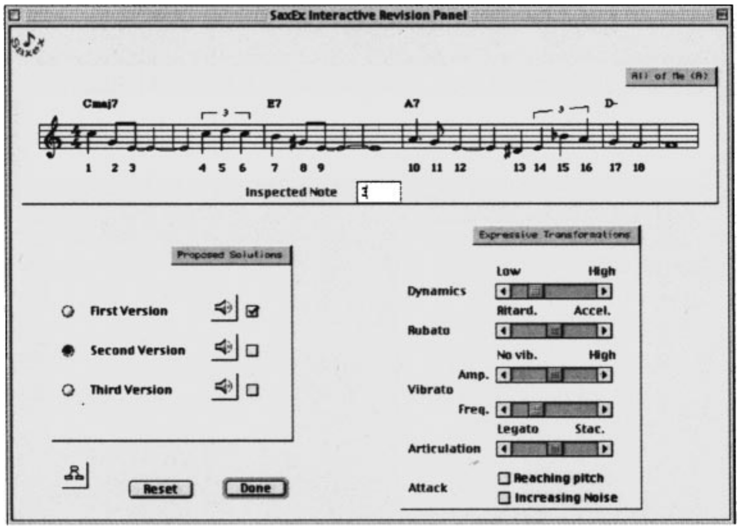

Figure 7. *SaxEx* panel for interactive revision and retention.

subpanels is to determine the criteria to be used and their combination for each task, respectively.

4.2.1. The Retrieval Panel. In this panel users can choose which perspectives have to be built by the *identify* subtask and which subtask will use them (the *search* or the *select* subtask). Moreover a combination partial order has to be specified.

Perspectives used by the *search* subtask will act as filters. Perspectives used by the *select* subtask will act only as a preference. Retrieval perspectives (described in Section 3.3) are grouped according to the musical model they come from (see Fig. 6): Affective knowledge, IR model, GTTM model, or Jazz and general music models.

User preferences for ranking the precedents found by the *search* subtask in the memory of cases is indicated by numbers. Perspectives with lower numbers have a higher preference. Perspectives with equal numbers represent no preference among them. Specifically, perspectives with equal numbers are combined using the `preference-union` *Noos* method and perspective with lower numbers are combined using the `hierarchical-preference-union` *Noos* method (see Section 2.2.2).

The default strategy of *SaxEx* is shown in Fig. 6 where only affective labels, IR role, and metrical strength are used by the *search* subtask—the remaining ones are used by the select subtask. The most preferred perspective is affective labels; the second is the IR role; then, metrical strength and harmonic stability are equally preferred; next, melodic duration, durational cumulation, and note duration are equally preferred; and finally, least preferred are time-span and prolongational-reduction.

4.2.2. The Reuse Panel. In this panel (see Fig. 6) users can choose which criteria to use for adapting solutions from cases to the current problem. Since some criteria are mutually exclusive, they cannot be activated at the same time. The order of preference among reuse criteria has to be a total order. That is, no number can be duplicated. Moreover, the random criterion is always active and discriminates among the last remaining alternatives.

In this panel users can also specify the number of solutions that will be provided by *SaxEx* just by typing a number from one to three (the default value).

4.3. Interactive Revision and Retention

The goal of the revision panel (see Fig. 7) is to allow the user to listen to proposed solutions, to inspect expressive transformations applied to each note, to revise them by means of proposing new values, to change retrieval and reuse criteria to obtain new expressive solutions, and to select solutions to be retained. The revision panel is organized into three subpanels: a subpanel on the top showing the score and for selecting a note to inspect and transform; a subpanel on the bottom left for listening to proposed solutions and selecting them for memorization; and a subpanel on the bottom right for revising the expressive transformations applied to each note.

First of all, the user can activate one of the proposed solutions by clicking in the radio buttons in the bottom left subpanel. Then, the user can listen to the selected solution and inspect the expressive transformations applied to each note by entering the note number in the score subpanel. When the note number changes, the expressive transformations subpanel shows the current values for the expressive transformations applied to that note. Then, the user can modify these values and listen to the modified expressive version.

From the revision panel, the user can go back to the customization panel (see Section 4.2 and Fig. 7) and perform experiments by changing the retrieval and reuse criteria. After the customization, the system goes back to the revision panel showing the set of new proposed solutions.

Finally, the user can select (see the bottom left subpanel in Fig. 7) which expressive solutions must be incorporated in the episodic memory of *SaxEx*. That is, the interactive retain subtask will influence the expressive solutions that *SaxEx* will propose according to the user's preferences.

5. System Evaluation

The set of experiments conducted with the interactive version of *SaxEx* focused on evaluating how the interactive capabilities of *SaxEx* allow the users to influence the results of the system according to their personal musical preferences and how these different results are perceived by them. The hypothesis being that the creative process involved in the generation of expressive music is influenced by these personal musical preferences. The strategy followed in evaluating the system was the following:

1. We selected two musical phrases belonging to the 'All of me' ballad and two musical phrases belonging to the 'Autumn leaves' ballad as input problems (four inexpressive phrases of about twenty notes), and ten different expressive performances of the 'How high the moon' ballad (having about twenty notes each).
2. We asked *SaxEx* to generate two versions of each musical phrase according to two different affective requests: Tender and Sad (T-S); Joyful and Restless (J-R). We obtained eight initial expressive interpretations.
3. We interactively changed parameters in the retrieve/reuse *SaxEx* panel (see Fig. 6). Specifically, we decreased the weight of harmonic stability, increased the weight of melodic duration, and used the minority rule in the 'Autumn leaves' phrases and the continuity and random rules in the 'All of me' phrases. We obtained another eight expressive interpretations.
4. Finally, using the interactive revision panel (see Fig. 7) we manually modified the way some of the notes were expressively transformed (for example increasing or decreasing the dynamics or the rubato). We obtained another eight expressive interpretations.

After *SaxEx* generated the twenty four expressive interpretations, we presented them—with the four inexpressive initial performances—to musical experts to evaluate the results. First of all, we requested two external experts (mentioned in the acknowledgements) to evaluate the differences in expressivity among each inexpressive version and its corresponding T-S version and J-R version (generated in step 2). Specifically, they assessed the degree for each affective dimension. Next, they evaluated the performances generated in step 3 and compare with those generated in step 2. Finally, they evaluated the differences perceived in performances from step 4.

Regarding the results of comparing the inexpressive initial phrases with their corresponding T-S and J-R versions generated in step 2, two main conclusions can be extracted: first, all the experts distinguished clearly the T-S and J-R expressive interpretations generated by *SaxEx*. The difference among them was that some experts assessed affective dimensions with higher values than others. This second result supported our hypothesis that the generation of expressive music is a creative process influenced by the personal preferences of each musician and, then, the interactive capabilities of the new version of *SaxEx* are strongly needed in order to generate expressive performances with a higher quality according to those personal preferences.

This hypothesis, which motivated the development of interactive tools for involving the user in the CBR process, was strengthened by the results of the evaluation of experts in comparing the expressive solutions generated in step 2 with those generated in step 3. All of them agreed in classifying the interpretations in the T-S and J-R affective space only introducing small variations and remarking not exactly the same influence of the expressive parameters (for instance one assessed the importance of vibrato while another emphasized the changes on dynamics). Since all those variations can be performed using the current interactive capabilities, the necessity of the interactive capabilities was enforced by the experts.

In the last evaluation step, comparing the differences perceived in the manual modifications performed using the interactive revision panel, all of them identified the differences. Moreover, and according to the personal perception expressed in the previous evaluation phase, they suggested to use the interactive revision panel for tuning the results proposed by *SaxEx* in different ways. Again, those tuning processes for improving the results produced by *SaxEx* are now possible because of the interactive capabilities of *SaxEx*.

6. Related Work

Previous work on interactive CBR has been addressed by [12] within the so called user-driven Conversational CBR systems. These systems iteratively interact with a user in a conversation to solve a query. The work in [12] focuses on the problem of revising case libraries according to case design guidelines in order to improve the conversational CBR performance.

In [13], the authors combine human and automated planners to interactively construct a plan in realistic and complex situations. This approach is similar to ours in the sense that the user can intervene basically in all the basic decision steps. That is, the interface provides mechanisms to save cases created generatively or analogically, to retrieve old cases (either manually or automatically) matching current situations and to choose various cases interleaving strategies for adaptation and replay.

Previous work on the analysis and synthesis of musical expression has addressed the study of at most two parameters such as rubato and vibrato [14, 15, 16], or rubato and articulation by means of an expert system [17]. Other work, such as in [18], is focused on the study of how a musician's expressive intentions influence the performances.

To the best of our knowledge, the only previous work addressing the issue of learning to generate expressive performances based on examples is that of Widmer [19], who uses explanation-based techniques to learn rules for dynamics and rubato in the context of a MIDI electronic piano. In our approach we deal with additional expressive parameters in the context of an expressively richer instrument because MIDI instruments have serious limitations regarding expressivity. Furthermore, this is the first attempt to deal with this problem using case-based techniques as well as the first attempt to cover the full cycle from an input sound file to an output sound file going in the middle through a symbolic reasoning and learning phase.

7. Conclusions and Future Work

In this paper we presented a new version of *SaxEx* where the case-based reasoner has been improved to allow the user to interact with the system during the CBR process. This capability was required because of two main reasons: we want to provide a tool for educational purposes—that is, with flexible experimentation capabilities; and the automatic generation of expressive music involves a creative process where user personal preferences cannot be fixed in advance. Specifically, regarding the educational use of the system, during the system evaluation the interactive capabilities introduced in *SaxEx* have been shown to be an important tool for learning how the five expressive parameters present in the system affect the resulting expressive solutions and also see the reasons why some expressive values, like for example a variation in dynamics (crescendo or diminiendo), that are well suited in some notes and not in others (like for example in notes of an ascending or descending melodic progression).

The new capabilities added to *SaxEx* have improved its usability, and we are planning several other capabilities. Concerning the retrieval subtask, we are thinking about how *SaxEx* can show the precedents selected for each note and the preference order among them. Moreover, it could be useful if the user was allowed to change that order interactively.

Concerning the reuse subtask, we are considering two different alternatives. The first one is to allow different reuse criteria for each expressive parameter. In the current version the reuse criteria is the same for all parameters but the user could take some risk on some expressive parameters (e.g., using minority rules) and be more conservative (e.g., using majority rules) in others.

The second alternative we are considering is to model the degree of the different expressive parameters by means of fuzzy sets, since they are closer than discrete labels to the continuous character of the SMS analysis. This change in the expressive model offers new interactive possibilities: on the one hand, the user will be able to manipulate fuzzy membership functions; on the other hand, more reuse criteria can arise. For instance, *SaxEx* will able to combine solutions provided by several cases using fuzzy combination operators such as those used in the defuzzyfication stage of fuzzy controllers [20].

Finally, concerning the retain subtask, we are planning to offer the user the possibility to select subphrases of solutions. At this moment the user is required to choose the entire solution but the user may prefer the first passage from one proposed solution and the second passage from another one. Moreover, in the current version the user can only provide positive feedback—i.e., selects only solutions that she likes. Another possibility is to provide negative feedback to the system in order to improve its future reasoning.

Acknowledgments

The research reported in this paper is partly supported by the ESPRIT LTR 25500-COMRIS *Co-Habited Mixed-Reality Information Spaces* project. We also acknowledge the support of ROLAND Electronics de España S.A. to our AI and Music project.

The authors acknowledge the collaboration of the sound modeling and processing group from the Pompeu Fabra University (and especially to Xavier Serra) on the SMS modules and the system evaluation. We also acknowledge Jordi Sabater the comments and discussions during the system evaluation.

The University of Padova has provided the excellent recordings of the 'How high the moon' ballad. We extend our thanks to David W. Aha and Héctor Múñoz-Avila for their comments for improving this paper.

References

1. W. Jay Dowling and Dane L. Harwood, *Music Cognition*, Academic Press, 1986.
2. Josep Lluís Arcos, Ramon López de Mántaras, and Xavier Serra, "Saxex: A case-based reasoning system for generating expressive musical performances," *Journal of New Music Research*, vol. 27, no. 3, pp. 194–210, 1998.
3. Agnar Aamodt and Enric Plaza, "Case-based reasoning: Foundational issues, methodological variations, and system approaches," *Artificial Intelligence Communications*, vol. 7, no. 1, pp. 39–59, 1994.
4. M. Elton, "Artificial creativity: Enculturing computers," *Leonardo*, vol. 28, no. 3, pp. 207–213, 1995.
5. Josep Lluís Arcos and Enric Plaza, "Noos: An integrated framework for problem solving and learning 7th workshop," in *Knowledge Engineering: Methods and Languages*, edited by Enriko Motta, Knowledge Media Institute, Open University, 1997.
6. Josep Lluís Arcos and Enric Plaza, "Inference and reflection in the object-centered representation language Noos," *Journal of Future Generation Computer Systems*, vol. 12, pp. 173–188, 1996.
7. Josep Lluís Arcos and Ramon López de Mántaras, "Perspectives: A declarative bias mechanism for case retrieval," in *Case-Based Reasoning. Research and Development*, edited by David Leake and Enric Plaza, Lecture Notes in Artificial Intelligence, Springer-Verlag, vol. 1266, pp. 279–290, 1997.
8. Enric Plaza, "Cases as terms: A feature term approach to the structured representation of cases," in *Case-Based Reasoning, ICCBR-95*, edited by Manuela Veloso and Agnar Aamodt, Lecture Notes in Artificial Intelligence, Springer-Verlag, vol. 1010, pp. 265–276, 1995.
9. Eugene Narmour, *The Analysis and Cognition of Basic Melodic Structures: The Implication-Realization Model*, University of Chicago Press, 1990.
10. Fred Lerdahl and Ray Jackendoff, "An overview of hierarchical structure in music," in *Machine Models of Music*, edited by Stephan M. Schwanaver and David A. Levitt, The MIT Press, pp. 289–312, 1993. Reproduced from Music Perception.
11. Josep Lluís Arcos, Dolores Cañamero, and Ramon López de Mántaras, "Affect-driven cbr to generate expressive music," in *Case-Based Reasoning. Research and Development. ICCBR'99*, edited by Karl Branting and Klaus-Dieter Althoff, Lecture Notes in Artificial Intelligence, Springer-Verlag, vol. 1650, pp. 1–13, 1999.
12. David W. Aha and Leonard A. Breslow, "Refining conversational libraries," in *Case-Based Reasoning. Research and Development*, edited by David Leake and Enric Plaza, Lecture Notes in Artificial Intelligence, Springer-Verlag, vol. 1266, pp. 267–278, 1997.
13. Michael T. Cox and Manuela M. Veloso, "Supporting combined human and machine planning: An interface for planning by analogical reasoning," in *Case-Based Reasoning. Research and Development*, edited by David Leake and Enric Plaza, Lecture Notes in Artificial Intelligence, Springer-Verlag, vol. 1266, pp. 531–540, 1997.
14. Manfred Clynes, "Microstructural musical linguistics: Composers' pulses are liked most by the best musicians," *Congition*, vol. 55, pp. 269–310, 1995.
15. P. Desain and H. Honing, "Computational models of beat indication: The rule-based approach," in *Proceedings of IJCAI'95 Workshop on AI and Music*, 1995, pp. 1–10.
16. H. Honings, "The vibrato problem, comparing two solutions," *Computer Music Jounal*, vol. 19, no. 3, pp. 32–49, 1995.
17. M.L. Johnson, "An expert system for the articulation of Bach fugues melodies," in *Readings in Computer-Generated Music*, edited by D.L. Baggi, IEEE Computes Society Press, pp. 41–51, 1992.
18. Giovani De Poli, Antonio Rodá, and Alvise Vidolin, "Note-by-note analysis of the influence of expressive intentions and musical structure in violin performance," *Journal of New Music Research*, vol. 27, no. 3, pp. 293–321, 1998.
19. Gerhard Widmer, "Learning expressive performance: The structure-level approach," *Journal of New Music Research*, vol. 25, no. 2, pp. 179–205, 1996.
20. Hamid R. Berenji, "Fuzzy logic controllers," in *An Introduccion*

to Fuzzy Logic Applications in Intelligent Systems, edited by Ronald R. Yager and Lofti A. Zadeh, Kluwer, pp. 69–96, 1992.

Josep Lluís Arcos is a Postdoctoral researcher of the Artificial Intellligence Institute of the Spanish Scientific Research Council (IIIA-CSIC). He received an M.S. on Musical Creation and Sound Technology from Pompeu Fabra Institute in 1996, and a Ph.D. in Computer Science from the Universitat Politecnica de Catalunya in 1997. He is the recipient of the "Swets & Zeitlinger Distinguished Paper Award" of the International Computer Music Association in 1997. He is working on Representation Languages for Integrating Problem Solving and Learning, on the Integration of Software Agents with Learning Capabilities in Information Spaces, and on Artificial Intelligence Applications to Music.

Dr. Ramon López de Mántaras is a Research Professor at the Artificial Intelligence Research Institute of the Spanish Council for Scientific Research. He holds an M.Sc. in Computer Science from the University of California at Berkeley, a Ph.D. in Physics from the University of Toulouse (France) and a Ph.D. in Computer Science from the Technical University of Barcelona (Spain). He has been an associate professor at the Technical University of Barcelona, a visiting professor at the University of Paris VI, member of the editorial board of several journals and former editor-in-chief of *AI Communications*. He is the recipient of the "City of Barcelona" research prize in 1982, the "Digital European AI Research Paper Award" in 1987, and the "Swets & Zeitlinger Distinguished Paper Award" of the International Computer Music Association in 1997, among other awards. He is presently working on Case-Based Reasoning, Autonomous Agents, and on AI Applications to Music.