Integrating background musical knowledge in a CBR system for generating expressive musical performances

Josep-Lluis Arcos[1], Ramon Lopez de Mantaras[1], Xavier Serra[2]

[1]IIIA, Artificial Intelligence Research Institute
CSIC, Spanish Scientific Research Council
arcos@iiia.csic.es, mantaras@iiia.csic.es
[2] IUA, Audiovisual Institute, Pompeu Fabra University
xserra@iua.upf.es

Abstract

This paper briefly describes a system called SaxEx, capable of generating expressive musical performances based on examples. We have done several recordings of a tenor sax expressively playing Jazz ballads. These recordings are analyzed, using spectral modelling techniques, to extract information related to five expressive parameters. The results of this analysis, together with the score, constitute the set of examples (cases) of the case-based component of SaxEx. From these examples, plus background musical knowledge based on Narmour's implication/realization theory of musical perception and Lerdahl and Jackendoff's generative theory of tonal music understanding (GTTM), SaxEx is able to infer a set of expressive transformations to apply to any given input sound file containing an inexpressive musical phrase of another ballad. Finally, SaxEx uses its spectral synthesis capabilities to actually apply the inferred transformations to the input sound file resulting in an expressive sound file at the output.

Introduction

SaxEx (Arcos, López de Mántaras, and Serra 1998) uses a CBR problem solver and background musical knowledge to infer a set of expressive transformations to be applied to every note of an inexpressive phrase given as input problem. 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 (spectral Modeling and Synthesis) (Serra 1997) sound analysis and synthesis techniques. The reasoning phase is performed using case-based techniques and is the main focus of this paper. SaxEx is implemented in Noos, a reflective object-centered representation language designed to support knowledge modeling of problem solving and learning. The first section of this paper briefly describes the role that SMS plays in the overall system. The second section provides some information about the object-centered language Noos that has been used to implement our system. In the third section we describe the overall SaxEx system and in particular how the background musical knowledge comes into play. Next we describe the experiments we have done with standard Jazz ballads and finally we present some conclusions and point to some further work.

Spectral Modeling and Synthesis

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 sounds 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 dynamics, rubato, vibrato, and articulation. The SMS synthesis procedure allows the generation of expressive reinterpretations by appropiately 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 any 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 original sound.

Noos

SaxEx is implemented in Noos (Arcos 1997), a reflective object-centered representation language designed to support knowledge modeling of problem solving and learning. Modeling a problem in SaxEx requires the integration of three different types of knowledge: domain knowledge, problem solving knowledge, and metalevel knowledge. Noos allows for a seemless integration of these three types of 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 by concepts such as notes, chords, Narmour's implication/realization structures (Narmour 1990), Lerdahl and Jackendoff's GTTM (Lerdahl and Jackendoff 1993), 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 the ways to solve 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 of subtasks and an specific combination of the results of the subtasks in order to solve the task it performs. For a given task there may be multiple alternative methods that may be capable of solving the task in different situations. This recursive decomposition of task into subtasks by means of a method is called the task/method decomposition.

The metalevel of Noos incorporates Preferences to model decision making about sets of alternatives present in domain knowledge and problem solving knowledge. 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.

Once a problem is solved, Noos automatically memorizes (stores and indexes) that problem. The collection of problems that a system has solved is called the Episodic memory of Noos. The problems solved by Noos are accessible and retrievable. This introspection capability of Noos is the basic building block for integrating learning, and specifically case-based reasoning, into Noos.

Noos also incorporates Perspectives (Arcos and López de Mántaras 1997), a mechanism to describe declarative biases for case retrieval in structured and complex representations of cases. Perspectives provide a flexible and dynamical way of retrieval in the episodic memory and are used by SaxEx for making decisions about the relevant aspects of a problem.

SaxEx

An input for SaxEx is a musical phrase described by means of its musical score (a MIDI file) and a sound. 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. The output of the system is a new sound file, obtained by transformations of the original sound containing an expressive performance of the same phrase. Solving a problem in SaxEx involves three phases: the analysis phase, the reasoning phase, and the synthesis phase. Analysis and synthesis phases are implemented using SMS sound analysis and synthesis techniques. The reasoning phase is performed using case-based techniques and implemented in Noos and is the main focus of this paper.

SaxEx has been developed specifying different types of knowledge: (1) modeling the concepts and structures relevant for representing musical knowledge, and (2) developing a problem solving method for inferring a sequence of expressive transformations for a given musical phrase. Problems to be solved by SaxEx are represented as complex structured cases embodying three different kinds of musical knowledge: (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 the pitch of the note (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. Moreover, a note holds, for example, the metrical-strength feature, inferred using GTTM theory, expressing the note's relative metrical importance into the phrase. The inference of this as well as other features is the main role played by the integrated background musical knowledge in the problem solving process of SaxEx. Chords hold also a set of features such as the name of the chord (Cmaj7, E7, etc), their position, their duration, and a reference to the next chord.

The musical analysis representation embodies structures of the phrase inferred using Narmour's and GTTM background musical knowledge. Narmour's implication/realization model (IR) 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.

On the other hand, Lerdahl and Jackendoff's generative theory of tonal music (GTTM) offers an additional approach to understanding melodies based on a hierarchical structure of musical cognition. GTTM proposes four types of hierarchical structures associated with a piece. This structural approach provides the system with a complementary view for inferring relevant aspects of melodies.

The information about the expressive performances contained in the examples of the case memory, is represented as a sequence of events extracted using the SMS sound analysis capabilities, therefore, this integrated spectral modelling component plays also a crucial role in providing additional reasoning power to the whole system. There is an event for each note within the phrase embodying information about expressive parameters applied to that note. Specifically, an event holds knowledge about dynamics, rubato, vibrato, articulation, and attack. These expressive parameters are described using qualitative labels as follows:

Changes on 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 on 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: the frequency vibrato level and the amplitude vibrato level. 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 can be introduced without altering the system.

The SaxEx task

The task of SaxEx is to infer, using a CBR problem solver and background musical knowledge, a set of expressive transformations to be applied to every note of an inexpressive phrase given as input problem. These transformations concern the dynamics, rubato, vibrato, articulation and attack of each note in the inexpressive phrase. For each note in the phrase, the following subtask decomposition is performed by the case-based problem solving method implemented in Noos:

Retrieve: The goal of the retrieve task is to choose, in the memory of cases (pieces played expressively) , the set of notes most similar to the current note problem. This task is decomposed in three subtasks:

1) Identify: The goal of this subtask is to build retrieval perspectives using the musical background knowledge integrated in the system. This gives two possible declarative retrieval biases: a first bias based on Narmour's implication/realization model, and a second bias based on Lerdahl and Jackendoff's generative theory. These perspectives guide the retrieval process by focusing it on the most relevant aspects of the current problem.

2) Search: the goal of this second subtask is to search cases in the case memory using Noos retrieval methods and some previously constructed Perspective(s). For instance let us assume that, by means of a Perspective, we declare that what makes two notes similar is the fact that they play the same role according to one of the criteria of Narmour's model. Then, the Search subtask will search for notes in the expressive performances that, following this Narmour's criterion, play the same role than the current problem note. One such criterion is, for example, that a note is the first note of an ascending or descending note progression.

3) Select: the goal of the select subtask is to rank the retrieved cases using Noos preference methods. The preference methods use criteria such as similarity in duration of notes, harmonic stability, or melodic directions.

Reuse: the goal of the reuse task is to choose, from the set of more similar notes previously selected, a set of expressive transformations to be applied to the current problem note. 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.

Retain: the incorporation of the new solved problem to the memory of cases is performed automatically in Noos. All solved problems will be available for the reasoning process in future problems.

Experiments

We are studying the issue of musical expression in the context of tenor saxophone interpretations. We have done several recordings of a tenor sax performer playing several Jazz standard ballads ('All of me', 'Autumn leaves', 'Misty', and 'My one and only love') with different degrees of expressiveness, including an inexpressive interpretation of each piece. These recordings are analyzed, using the SMS spectral modeling techniques, in order to extract basic information related to the expressive parameters. The set of extracted parameters together with the scores of the pieces constitute the set of structured cases of the case-based system. From this set of cases and using similarity criteria based on background musical knowledge, the system infers a set of candidate expressive transformations for a given inexpressive piece. Finally, using the set of inferred transformations and the SMS synthesis procedure, SaxEx generates an expressive reinterpretation of the inexpressive piece.

We have performed two sets of experiments combining the different Jazz ballads recorded. The first set of experiments consisted in using examples of three different expressive performances of twenty note phrases of a piece in order to generate an expressive reinterpretation of another inexpressive phrase of the same piece. This group of experiments has revealed that SaxEx identifies clearly the relevant cases even though the new phrase introduces small variations with respect to the phrases existing in the memory of precedent cases.

The second set of experiments consisted in using examples of expressive performances of some pieces in order to generate expressive reinterpretations of different inexpressive pieces. More concretely, we have worked with three different expressive performances of a piece having about fifty notes in order to generate expressive reinterpretations of twenty-note inexpressive phrases of a different piece. This second group of experiments has revealed that the use of perspectives in the retrieval step allows to identify situations such as long notes, ascending or descending melodic lines, etc. Such situations are also usually identified by a human performer.

Conclusions and future work

To the best of our knowledge, this is the first attempt to deal with the problem of generating expressive musical performances 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. The results obtained are comparable to a human performance specially for dynamics, rubato and vibrato, however the articulation and attack need further work.

Concerning additional future work, we also intend 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.

model the decay of long notes by means of different envelope functions decreasing more or less rapidly.

experiment further with different expressive parameters and their different degrees of expressiveness.

With the aim of making our system useful for musicians we intend to provide the possibility of interactive revision of the proposed solutions by the user. In this way the user will have the possibility to filter those solutions that should be retained. This capability will allow the user to tailor the system according to his preferences.

Integrating different types of knowledge including spectral modeling techniques as well as the appropriate background knowledge was the key to the success of our system and we believe that, in general, background knowledge is essential when dealing with real world problems that require the representation of complex structured cases. In our system, this background knowledge proved to be very useful to guide the retrieval of relevant cases by means of Perspectives.

References

Arcos, J.L 1997. The Noos representation language. Phd. diss., Dept. of Computer Science, Universitat Poliècnica de Catalunya, Barcelona, Spain.

Arcos, J.L.; and López de Mántaras, R. 1997. Perspectives: a declarative bias mechanism for case retrieval. In Leake, D., and Plaza, E. eds. Proceedings of the Second International Conference on Case Based Reasoning, *Lecture Notes in Artificial Intelligence* 1266:279-290.

Arcos, J.L.; López de Mántaras, R.; and Serra, X. 1998. SaxEx: a case-based reasoning system for generating expressive musical performances. *Journal of New Music Research* . Forthcoming.

Lerdahl, F.; and Jackendoff, R. 1993. An overview of hierarchical structure in music. In Schwanaver, S.M., and Levitt, D.A. eds. *Machine Models of Music*. The MIT Press.

Narmour, E. 1990. *The Analysis and cognition of basic melodic structures : the implication-realization model..* University of Chicago Press.

Serra, X. 1997. Musical sound modelling with sinusoids plus noise. In De Poli, G.D.; Picialli, A.; Pope, S.T.; and Roads, C. eds. *Musical Signal Processing*. Swets and Zeitlinger Publishers.

Appendix

1. Integration name/category: SaxEx
2. Performance Task: Generating expressive musical performances
3. Integration Objective: Case retrieval
4. Reasoning Components: CBR, Spectral Modeling Techniques (SMS)

5. Control Architecture: Sequential (CBR follows SMS analysis and is followed by SMS synthesis)
6. CBR Cycle Step(s) Supported: Pre-processing, retrieval, reuse, retention, post-processing
7. Representations: object-centered
8. Additional Reasoning Components: Narmour's implication/realization theory of musical perception, Lerdahl and Jackendoff's generative theory of tonal music
9. Integration Status: Empirical evaluation
10. Priority future work: Developping a useful application