Proc. the Eighteenth National Conference on Artificial Intelligence
(AAAI-02), pp.193-198, The AAAI Press / The MIT Press (2002)

Constructive Adaptive User Interfaces — Composing Music Based on Human Feelings

Masayuki Numao, Shoichi Takagi, and **Keisuke Nakamura**
Department of Computer Science, Tokyo Institute of Technology
2-12-1 O-okayama, Meguro-ku, Tokyo 152-8552, Japan
numao@cs.titech.ac.jp

Abstract

We propose a method to locate relations and constraints between a music score and its impressions, by which we show that machine learning techniques may provide a powerful tool for composing music and analyzing human feelings. We examine its generality by modifying some arrangements to provide the subjects with a specified impression. This paper introduces some user interfaces, which are capable of predicting feelings and creating new objects based on seed structures, such as spectra and their transition for sounds that have been extracted and are perceived as favorable by the test subject.

Introduction

Music is a flow of information among its composer, player and audience. A composer writes a score that players play to create a sound to be listened by its audience as shown in Figure 1. Since a score, a performance or MIDI data denotes a section of the flow, we can know a feeling caused by a piece of score or performance. A feeling consists of a very complex elements, which depend on each person, and are affected by a historical situation. Therefore, rather than clarifying what a human feeling is, we would like to clarify only musical structures that cause a specific feeling. Based on such structures, the authors constructed an automatic arrangement and composition system producing a piece causing a specified feeling on a person.

The system first collects person's feelings for some pieces, based on which it extracts a common musical structure causing a specific feeling. It arranges an existing song or composes a new piece to fit such a structure causing a specified feeling. In the following sections, we describe how to extract a musical structure, some methods for arrangement or composition, and the results of experiments.

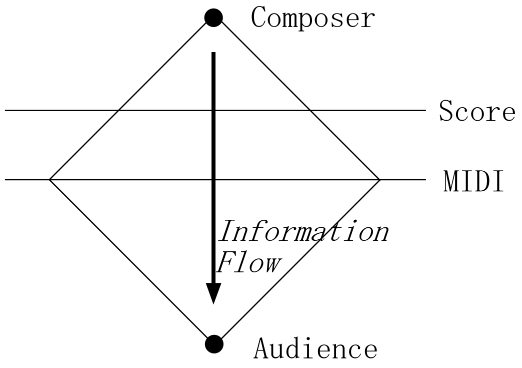


Figure 1: Information flow and authoring

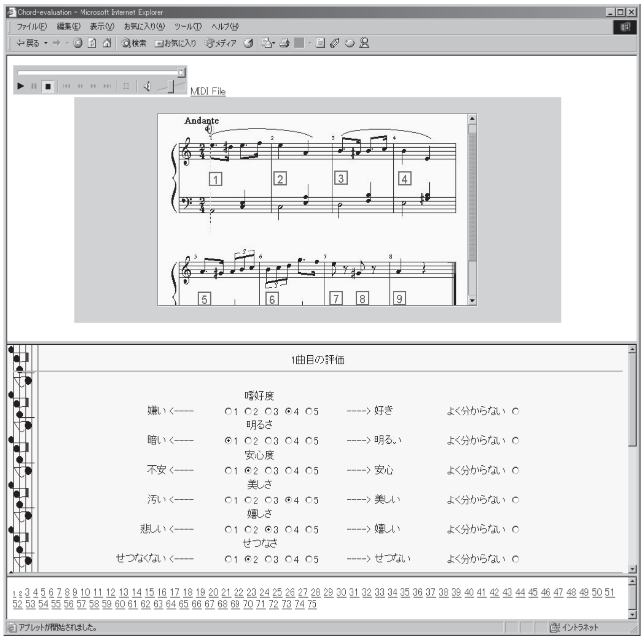
Extracting a musical structure

The system collects evaluation of some pieces in 5 grades for some adjective pairs via a web page as shown in Figure 2. The subject selects a music piece from the bottom menu containing 75 pieces, and evaluates it. The upper part is a MIDI player and a score. As well as the whole piece, it collects evaluation of each bar identified by $\boxed{1}$, $\boxed{2}$, ..., $\boxed{9}$. The middle part is a form to input evaluations, where the adjective pairs are written in Japanese.

To extract a structure that affects a feeling, the system analyzes some scores based on the theory of tonal music, i.e., ones with tonality, cadence, borrow chord structures, etc. For example, it automatically extracts rules to assign a chord to each function, or from two or three successive functions (Numao, Takagi, & Nakamura 2002). By using inductive logic programming — a machine learning method to find rules written in the programming language *PROLOG*, it is possible to find such a structure based on background knowledge, such as the theory of tonal music. Its procedure is as follows:

1. By using Osgood's semantic differential method in psychology, each subject evaluates 75 pieces by 6 adjective pairs[1], each of which is in 5 grades.
2. Find a condition to satisfy each adjective by using a machine learning method based on inductive logic programming. For the first stage, positive examples are structures in pieces whose evaluation is higher than or equal

[1] (favorable, unfavorable), (bright, dark), (stable, unstable), (beautiful, ugly), (happy, unhappy), (heartrending, no heartrending).



1曲目の評価

嗜好度
嫌い <---- ○1 ○2 ○3 ●4 ○5 ----> 好き よく分からない ○
明るさ
暗い <---- ●1 ○2 ○3 ○4 ○5 ----> 明るい よく分からない ○
安心度
不安 <---- ○1 ●2 ○3 ○4 ○5 ----> 安心 よく分からない ○
美しさ
汚い <---- ○1 ○2 ○3 ●4 ○5 ----> 美しい よく分からない ○
嬉しさ
悲しい <---- ○1 ○2 ●3 ○4 ○5 ----> 嬉しい よく分からない ○
せつなさ
せつなくない <---- ○1 ●2 ○3 ○4 ○5 ----> せつない よく分からない ○

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75

Figure 2: Gathering Evaluation

to 5. Other structures are negative examples. This gives a generalized structure whose evaluation is better than 5 by each adjective pair. This condition earns 5 points for the adjective pair.

3. Similarly, find a condition to accomplish evaluation better than 4. This condition earns 4 points.

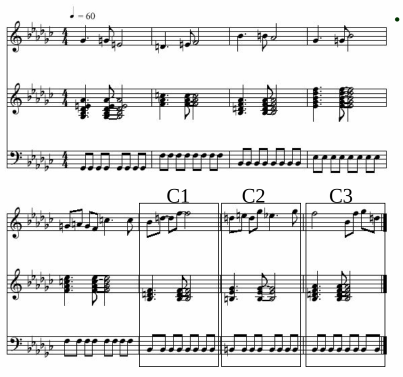
The condition for the opposite adjective, such as *dark*, *unfavorable* and *unstable*, earns $6 - g$ points, where g is the grade given by the user. Since 75 pieces are too many to be evaluated in one session, the subjects evaluate them in multiple sessions by comparing a pair of some chosen pieces multiple times.

Each rule is described by a predicate rather than an attribute, since it is hard to describe a score by using only some attributes. PROLOG describes each condition, whose predicates are defined in background knowledge (Numao, Kobayashi, & Sakaniwa 1997). We prepare the following predicates in PROLOG to describe a musical structure, where `frame` is the name of predicate and `/1` is the number of arguments:

1. `frame/1` represents the whole framework of music, i.e., tonality, rhythm and instruments.
2. `pair/2` represents a pattern of two successive chords.
3. `triplet/3` represents a pattern of three successive chords.

For example, we can describe that a subject likes a piece whose tonality is E major or E minor, tempo is Allegretto, accompanying instrument is piano, rhythm is 4/4, and contains a specified pair of successive chords.

To acquire such conditions, we use Inductive Logic Programming (ILP), which is a machine learning method to find a PROLOG program. A score is represented by a symbol, where a relation between two notes are important. These mean that ILP is a good tool for generalizing a score. Figure 3 shows a score and its generalization described in PROLOG. The variables `C1`, `C2` and `C3` represent successive bars. These clauses mean that *SubjectA* feels a piece dark when its tonality is moll (minor), its tempo is larghetto, the first chord is moll V, the second is triad (`form_V`) VI, and the third is 7th root position (`inversion_Zero`) chord.



- SubjectA dark
  - frame(S) :-
    tonality_moll(S),
    tempo_larghetto(S).
  - triplet(C1, C2, C3) :-
    moll(C1),
    form_V(C2),
    chord_VI(C2),
    chord_V(C1),
    inversion_Zero(C3),
    form_VII(C3).

Figure 3: Acquiring predicates.

Arrangement

The authors constructed the arranger and the composer separately, since arrangement is easier than composition, i.e., the composer is much slower than the arranger. The following method arranges a piece by minimally changing its chord sequence to cause the required feeling:

1. Analyze the original chords to recognize their function, e.g., tonic, dominant, subdominant, etc.
2. Modify each chord to satisfy the acquired conditions without changing its function.
3. Modify the original melody minimally to fit the modified sequence of chords.

This is accomplished by the following windowing procedure:

1. Set a window on the first three chords.
2. Enumerate the all chords with the same function to satisfy the acquired predicates `pair` and `triplet`. Sum up the points of acquired predicates to evaluate each chord sequence.
3. Shift the window by two, i.e., set a new window on the last chord and its two successors. Enumerate the chords similarly to the above.
4. Repeat the above to find a sequence with the most points.
5. Repeat the above for the all 12 tonality. Determine the tonality that earns the most points.
6. Determine the frame that earns the most points.

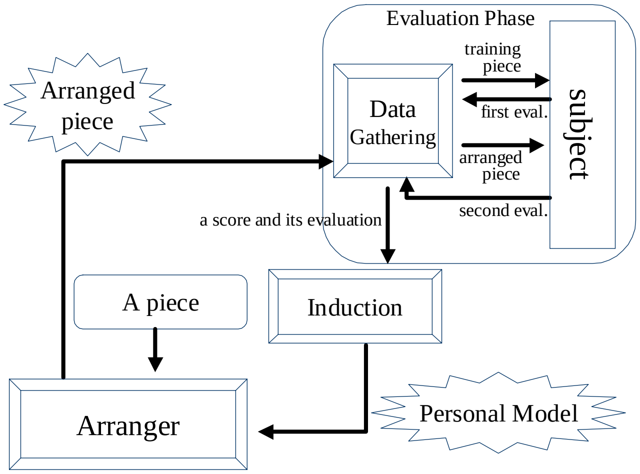


Figure 4: Arranger

Significance level	bright	stable	favorite	happy	beautiful	heartrending	
5 %	Yes	Yes	No	No	No	Yes	+
1 %	Yes	No	No	No	No	Yes	
5 %	Yes	Yes	Yes	No	Yes	No	-
1 %	Yes	No	Yes	No	Yes	No	

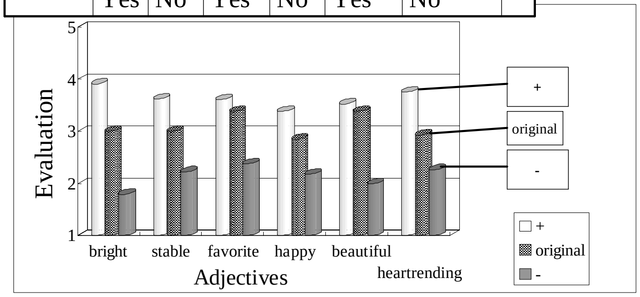


Figure 5: Evaluation of arrangements

The authors prepared 75 well-known music pieces without modulation[2], from which they extracted 8 or 16 successive bars. For automatic arrangement they prepared other three pieces. The flow of experiment is shown in Figure 4. The subject evaluated each piece as one of 5 grades for 6 pairs of adjectives: *bright - dark, stable - unstable, favorable - unfavorable, beautiful - ugly, happy - unhappy, heartrending - no heartrending*. For each adjective pair the system constructed a personal model of feeling, based on which it tried to arrange the prepared three pieces into ones causing a specified feeling, which were evaluated by the same subject.

The system was supplied 3 original pieces, and alternatively specified 6 adjective pairs, i.e., 12 adjectives. Therefore, it produced $3*12 = 36$ arranged pieces, whose average evaluation by the subjects is shown in Figure 5. In the figure, '+' denotes a *positive arrangement (composition)*, which is a bright, stable, favorable, beautiful, happy or heartrending arrangement (composition). '-' denotes a *negative arrangement (composition)*, which is the opposite: dark, unstable, unfavorable, ugly, unhappy, no heartrending. The results show that the positive arrangements resulted in higher evaluation, and that the negative arrangements resulted in lower evaluation for all the adjective pairs. According to the table in Figure 5, many of the results are statistically significant.

After the experiments in (Numao, Kobayashi, & Sakaniwa 1997), the system has been improved in collecting evaluation of each bar, introducing `triplet/3` and `frame/1`, and the search mechanism for chord progression. The above results support their effects.

Composition

Based on a collection of conditions ILP derives, we have obtained a personal model to evaluate a chord progression. A genetic algorithm (GA) produces a chord progression by using the model for its fitness function. Such a chord progression utilizes a melody generator to compose a piece from scratch rather than to arrange a given piece. The procedure to compose music based on a personal feeling is described in Figure 6. The subject evaluates each piece as one of 5 grades for the 6 pairs of adjectives. The ILP system finds relations between a set of score features and its evaluation, which are described by the predicates defined in background knowledge. These relations describe a feeling, based on which a genetic algorithm produces a chord progression.

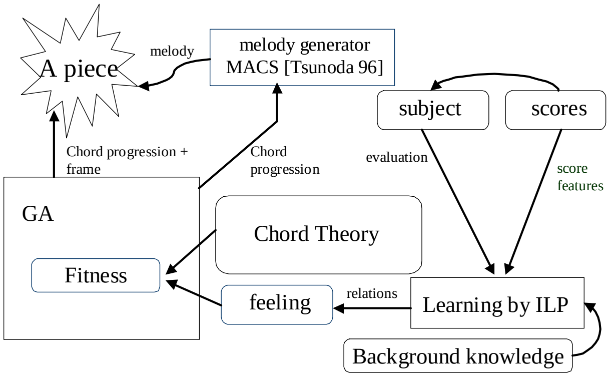


Figure 6: Composing system

A genotype, operators and a fitness function are important in genetic algorithms. Figure 7 shows the genotype for producing a chord progression. To represent complicated parameters, a bit string in GA is extended to a matrix, where a bit is extended to a column in the matrix. Therefore, the crossover operator splits and exchanges a string of columns. The fitness function reflects a music theory and the personal model:

$$\begin{aligned} &Fitness\_Function(M) = \\ &\quad Fitness\_Builtin(M) + Fitness\_User(M) \end{aligned}$$

where M is a score described by a predicate `music/2`.

[2] 39 Japanese *JPOP* songs and 36 pieces from classic music or textbooks for harmonics.

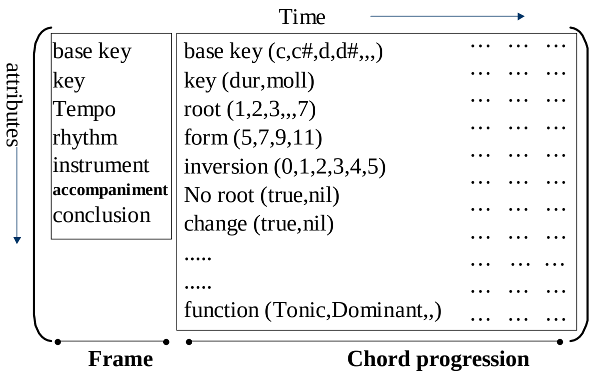


Figure 7: Genotype

This makes possible to produce a chord progression that fits the theory and causes the required feeling. $Fitness\_Builtin(M)$ is a fitness function based on the theory of tonal music, which issues a penalty to a chord progression violating the theory. $Fitness\_User(M)$ is based on the extracted musical structures that reflect the subject's feelings:

$$\begin{aligned} Fitness\_User(M) &= Fitness\_Frame(M) \\ &\quad +Fitness\_Pair(M) \\ &\quad +Fitness\_Triplet(M) \end{aligned}$$

where $Fitness\_Frame(M)$ is fitness based on tonality, rhythm and instruments, etc. $Fitness\_Pair(M)$ and $Fitness\_Triplet(M)$ are based on two or three successive chords, respectively.

For producing a piece, the system uses MACS (Tsunoda 1996), which generates a melody from a chord progression and some rules for the duration. Since MACS is a black box containing complicated program codes, the authors start a new project to find simple rules describing the process, which clarifies the process of generating a melody.

Figure 8 and 9 show created pieces. Figure 8 is a piece the system tried to make bright. Figure 9 is one it tried to make dark. These examples show that the system composes a bright piece without handcrafted background knowledge on brightness and by automatically acquiring some musical structures that cause a bright feeling. Other created pieces are shown in (Numao, Takagi, & Nakamura 2002).

Figure 10 shows evaluation of the composed pieces. '+' shows the average result of pieces the system tried to make positive. '-' shows that it tried to make negative. According to Student's t-test, they are different for 4 adjective pairs at the level of significance $\alpha = 0.05$. They are different for 2 pairs at the level $\alpha = 0.01$. Figure 11 shows the effect of melody, which is dramatic in some adjective pairs.

This system is profoundly different from other composing systems in that it composes based on a personal model extracted from a subject by using a machine learning method. A composing system using an interactive genetic algorithm (IGA), such as GenJam (Biles 2002), may be similar method to ours in that it creates a piece based on the user interaction.

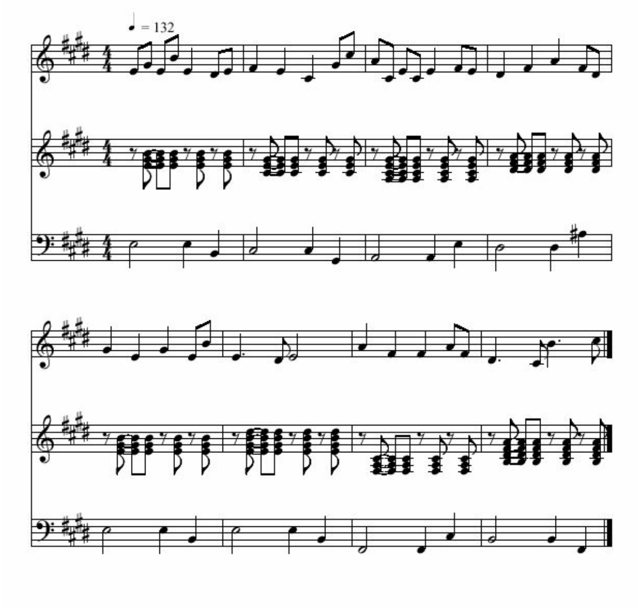


Figure 8: A created bright piece

However, IGA generally requires far more interactions than ours, which reduces the number of interactions by utilizing a personal model generalized from examples, although the detailed comparison between GenJam and ours is a future work. Other advantages are that we can recycle a personal model in many compositions, and manually tailor a predicate in the system to improve its performance.

Related Work

In algorithmic music composition, a simple technique involves selecting notes sequentially according to a transition table that specifies the probability of the next note as a function of the previous context. Mozer (1994) proposed an extension of this transition table approach using a recurrent autopredictive connectionist network. Our system is more flexible than this in that the user specifies an adjective to change impressions of a created piece.

Wiggins (1999) proposed to apply genetic algorithms to music composition. Our method combines a genetic algorithm with a personal model acquired by machine learning.

Widmer (1994) proposed a method of accomplishing explanation-based learning by attaching harmonies — chord symbols to the notes of a melody. The present paper further discusses a means of controlling the process based on learned feelings.

Hirata (1999, 1996) constructed a reharmonizing and arranging system based on a knowledge representation in Deductive Object-Oriented Databases (DOOD). Our system is different in adaptation mechanism by acquiring a personal model.

Thom (2000) proposed to apply unsupervised learning to interactive Jazz/Blues improvisation. In contrast, our method is an application of inductive learning, i.e., supervised learning. Hörnell's system produces and harmonizes

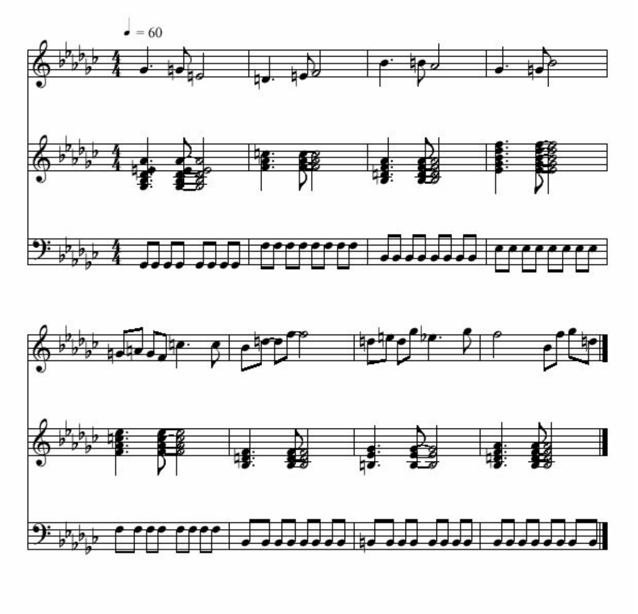


Figure 9: A created dark piece

simple folk style melodies based on learned musical structure (Hörnel & Ragg 1996). Dannenberg, Thom and Watson (1997) apply machine learning techniques to musical style recognition. Our method is different from them in its emotional-driven generation of music.

The Wolfgang system utilizes emotions to enable learning to compose music (Riecken 1998). It is an interesting research topic to compare its cultural grammar and our PROLOG rules based on the semantic differential method. Emotional coloring (Bresin 2000) is an interesting research in the field of automatic music performance with a special focus on piano, although automatic composition is out of its scope.

Conclusion

Pat Langley (1998) proposed an adaptive user interface to be applied to a navigation system (Rogers, Fiechter, & Langley 1999). Our method extends the concept of adaptive user interfaces in a sense that it constructs a new description adaptively. That is why we call our system a *constructive* adaptive user interface.

Acknowledgements

The authors would like to thank Pat Langley and Dan Shapiro, who gave fruitful comments, when one of the authors gave a talk at Center for the Study of Language and Information, Stanford University.

References

Biles, J. A. 2002. Genjam. http://www.it.rit.edu/˜ jab/GenJam.html.

Bresin, R. 2000. *Virtual Virtuosity*. Ph.D. Dissertation, Kungl Tekniska Högskolan, Stockholm.

Significance level	bright	stable	favorable	happy	beautiful	heartrending
5%	Yes	Yes	Yes	No	No	Yes
1%	Yes	No	No	No	No	Yes

Student's T-test

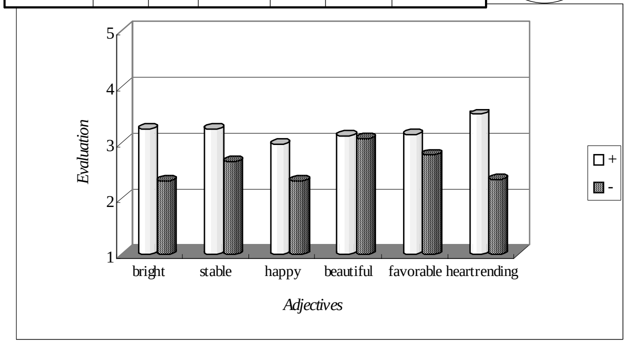


Figure 10: Evaluation of Composition

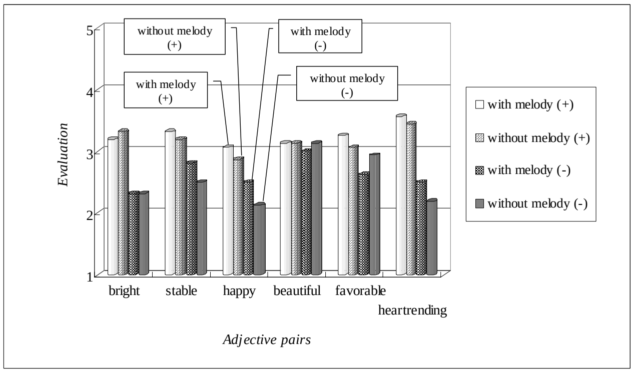


Figure 11: Effects of melodies

Dannenberg, R. B.; Thom, B. T.; and Watson, D. 1997. A machine learning approach to musical style recognition. In *Proc. ICMC97*.

Hirata, K., and Aoyagi, T. 1999. Musically intelligent agent for composition and interactive performance. In *Proc. ICMC*, 167–170.

Hirata, K. 1996. Representation of jazz piano knowledge using a deductive object-oriented approach. In *Proc. ICMC*.

Hörnel, D., and Ragg, T. 1996. Learning musical structure and style by recognition, prediction and evoloution. In *Proc. ICMC*. International Computer Music Association.

Langley, P. 1998. Machine learning for adaptive user interfaces. In *CSLI-Stanford University IAP Spring Tutorials*, 155–164.

Michalski, R. S., and Tecuci, G., eds. 1994. *Machine Learning: A Multistrategy Approach (Vol. IV)*. San Francisco, CA: Morgan Kaufmann.

Mozer, M. 1994. Neural network music composition by prediction: Exploring the benefits of psychoacoustic constraints and multiscale processing. *Connection Science*.

Numao, M.; Kobayashi, M.; and Sakaniwa, K. 1997. Acquisition of human feelings in music arrangement. In *Proc. IJCAI 97*, 268–273. Morgan Kaufmann.

Numao, M.; Takagi, S.; and Nakamura, K. 2002. CAUI demonstration — composing music based on human feelings. In *Proc. AAAI 2002*. AAAI Press.

Riecken, D. 1998. Wolfgang: “emotions” and architecture which enable learning to compose music. SAB’98 Workshop on Graounding Emotions in Adaptive Systems. http://www.ai.univie.ac.at/ paolo/conf/sab98/sab98sub.html.

Rogers, S.; Fiechter, C.-N.; and Langley, P. 1999. An adaptive interactive agent for route advice. In *Proc. the Third International Conference on Autonomous Agents*, 198–205.

Thom, B. 2000. Unsupervised learning and interactive Jazz/Blues improvisation. In *AAAI/IAAI*, 652–657.

Tsunoda, K. 1996. Computer-supported composition of music. Master Thesis, University of Mie.

Widmer, G. 1994. Learning with a qualitative domain theory by means of plausible explanations. In *(Michalski & Tecuci 1994)*. chapter 25, 635–655.

Wiggins, G., et al. 1999. Evolutionary methods for musical composition. *International Journal of Computing Anticipatory Systems*.