jPop-E: An Assistant System for Performance Rendering of Ensemble Music

Mitsuyo Hashida
School of Science & Technology
Kwansei Gakuin University
Sanda, 669-1337 JAPAN
hashida@ksc.kwansei.ac.jp

Noriko Nagata
School of Science & Technology
Kwansei Gakuin University
Sanda, 669-1337 JAPAN
nagata@ksc.kwansei.ac.jp

Haruhiro Katayose
School of Science & Technology
Kwansei Gakuin University
Sanda, 669-1337 JAPAN
katayose@ksc.kwansei.ac.jp

ABSTRACT

This paper introduces jPop-E (java-based PolyPhrase Ensemble), an assistant system for the Pop-E performance rendering system. Using this assistant system, MIDI data including expressive tempo changes or velocity control can be created based on the user's musical intention. Pop-E (PolyPhrase Ensemble) is one of the few machine systems devoted to creating expressive musical performances that can deal with the structure of polyphonic music and the user's interpretation of the music. A well-designed graphical user interface is required to make full use of the potential ability of Pop-E. In this paper, we discuss the necessary elements of the user interface for Pop-E, and describe the implemented system, jPop-E.

Keywords

Performance Rendering, User Interface, Ensemble Music Expression

1. INTRODUCTION

Performance rendering is one of the main topics of recent studies on music and computational technology [1, 2, 3]. One of the primary aims of performance rendering is to make a system generate a natural musical expression automatically. For this goal, performance rendering systems must be able to deal with abstract representations of delicate music expressions, and to select a unique interpretation among the various possible musical interpretations of the same piece. These requirements have not been met yet, and currently available performance rendering systems require manual operations to render music.

Interest in performance rendering is shifting to the expression of polyphonic music. Some interactive systems [4, 5] achieve this goal, but they require an expressive performance template for each piece, and it is difficult to polish the specified part of the performance due to their realtime processing.

We want to provide a simple interface that facilitates the design of polyphonic music expression. In this paper, we describe a performance design interface jPop-E implemented in Java; it is based on Pop-E [6], a rule-based performance rendering model for multi-part music that has won awards in listening contests of performance rendering[1].

2. EXPRESSION MODEL FOR POLYPHINC MUSIC

Pop-E (Polyphrase Ensemble) [6] is a rule-based performance expression model to render a natural expression of multi-part music. To realize a natural performance of ensemble music, it should satisfy to give an independent phrasing expression to every part and to align several timings of sound between parts accordingly. To satisfy these requirements, in Pop-E, every part is given an independent group structure and two categories of performance expression rules are applied to them. This procedure makes a temporal gap in the occupancy time which is necessary for a performance consisting of more than one part. This problem is resolved by synchronization between parts to align the onset time of each side of groups in parts to that of an *attentive part* (defined below).

Figure 1 illustrates the outline of Pop-E. A musical structure is given by a user, as an object to apply performance rules. User inputs include **the group structures of the parts** (pairs of notes that are the beginning and the last of a group), **the top notes of the groups** for the phrase expression which shapes a linear mountain, **an attentive part** that is a primary note sequence of a piece, and the control parameters of the rules. A part is assumed to consist of a single melody. The group structure of each part is used for applying all the phrase expression rules of a group and extracting the candidate synchronization positions between parts. The attentive part is utilized as criteria of the synchronous control and for applying the note stretching rules.

The performance rules of Pop-E are applied to each component without specifying the musical role (melody, base, accompaniment, etc.).

2.1 Performance Rules

The preset rules of the Pop-E model are classified into two categories: (1) expression of groups and (2) temporal note stretching (Agogik). The expression of the group structure

[1] Sound samples are available from
http://www.m-use.net/research/PopE/index-e.htm

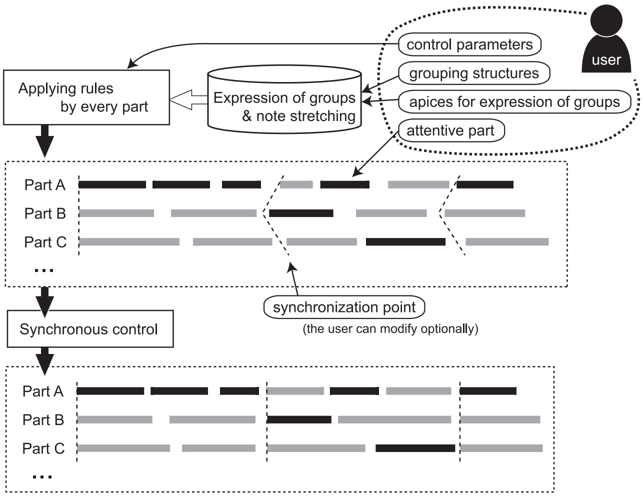

Figure 1: Concept of Pop-E

is a fundamental issue in musical performance.

Pop-E distinguishes between two typical group expressions:

(a) Giving an accent of velocity to the beginning note of a group.

(b) Phrase expression. This expression increases the dynamics and tempo from the beginning note to the top note of a group, after which it decreases them to the last note of that group to become the climax of that expression at the top note.

Another important factor for expressive performances is temporal note stretching: a small change in tempo or a mute or expansion of a sounding note. We are concerned with stretching the length of a note. Our model deals with the following notes.

(a) Grace notes and k-level triplets. The ratios of the initial values of velocity and length to all notes to accord are given. K-level triplets are used when notes appear irregularly rather than periodically as a rhythm pattern.

(b) Leaping notes. When the interval of pitch between two adjacent notes in the same part is larger, the ratios of the initial velocity and length values to other values are given to the precedent note.

(c) Attention transition notes. When the attentive part steps over other parts, the ratios of initial values of velocity and length to which other values are given to notes on the boundary.

2.2 Synchronization between Parts

Applying rules causes an unexpected timing lag between some parts in the same area of the score; the performance would not make any sense as music if that lag were neglected. To solve this problem and keep the individual expression of the parts, we need to estimate synchronization points to align parts. We then have to schedule the timing of all parts according to the timing of the attentive part. The synchronization points are identified by comparing the group structures given to each part. These points indicate the onset times of the beginning or the last note in a group. The notes

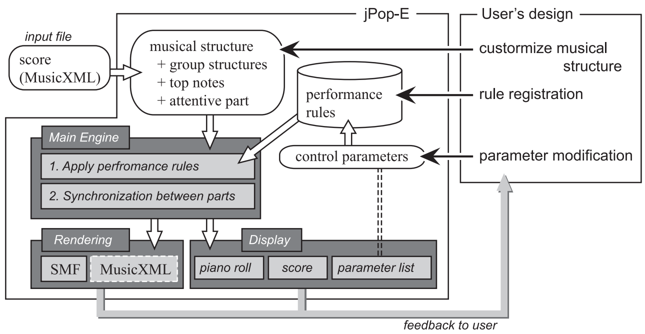

Figure 2: Outline of jPop-E

at the synchronization point sound at the same time. In the area between adjacent synchronization points, the relevant non-attentive parts are scaled linearly while maintaining the ratio of the note lengths.

3. jPop-E

Figure 2 shows the outline of "jPop-E." The input data for jPop-E are a music score file and a list of performance rules consisting of conditional and control parameters. The score file written in MusicXML format contains data on the pitch and duration of every note and structure information (**group structures of parts, top notes of groups and an attentive part**). The score file can be obtained from commercial notation software such as Finale by using the exporting function. The structural information is given using jPop-E functions (see section 3.3), or pre-edited using the articulation tool of the notation software when the user inputs note data. The system applies performance rules to the given score and synchronization between parts, and then renders an expressive performance, which can be viewed with a piano roll display. The system also allows the user to register new rules.

The user's concrete task is to modify some of the musical structure and control parameters. He or she repeats the cycle of adjusting parameters and listening to the rendering result until (s)he obtains the desired result.

jPop-E provides a GUI that assists with the iterative adjustment of the musical structures and control parameters. jPop-E is implemented on Java, considering compatibility with any OS, and the existence of a general MusicXML parser, and system extensibility.

3.1 Basic Operation

The system provides basic user operation functions for (1) rendering a performance (MIDI data generation), (2) listening to the rendered performance, (3) adjusting the control parameters (see section 3.2), and (4) editing the musical structures (see section 3.3). The user can issue commands regarding (1), (2) and (4) from the main display shown in Figure 3.

First, the user indicates a score file and a parameter list to be read. Then the system shows the note sequence of the score file on the piano roll. The user can view the parameter list in another window, by selecting the command from the menu, and can adjust each parameter. When the user pushes the 'render' command on the main display, the system starts applying performance rules using the current

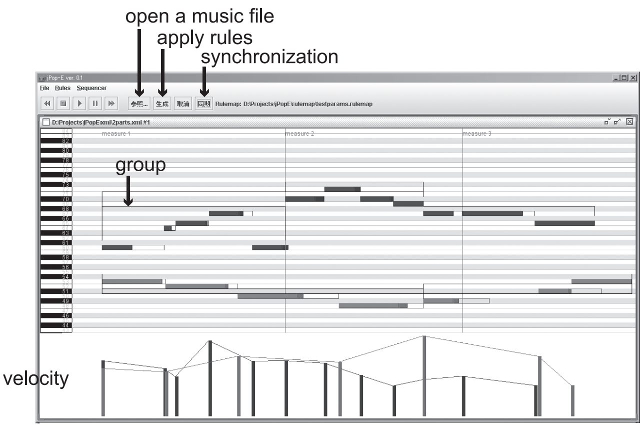

Figure 3: Main display of jPop-E

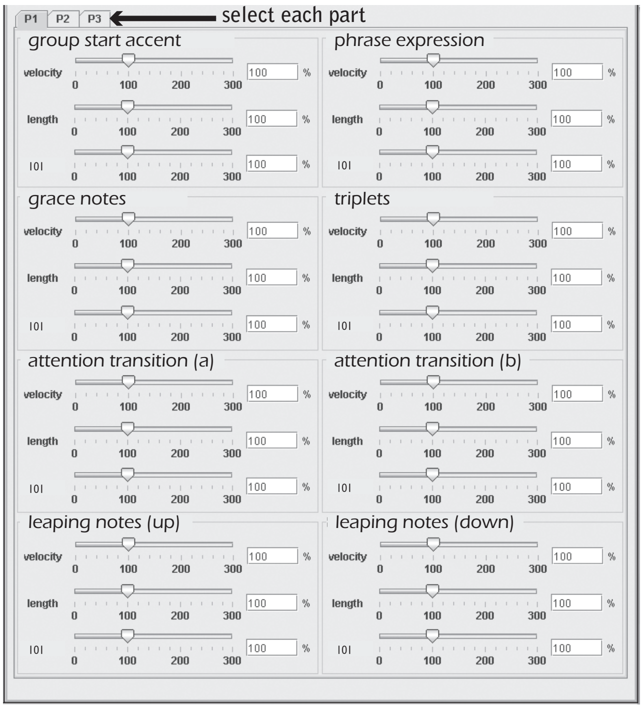

Figure 4: Control parameters display of the preset rules.

parameters, and the piano roll is refreshed to the latest performance. The user can choose whether the system should execute the synchronization procedure or not. To listen to the note sequence, a toolbar including 'play', 'stop', and 'pause' buttons is provided on the main display. The user can edit the group structure, if necessary.

3.2 Adjustment of Control Parameters

Control parameters regard the loudness and duration of each note and the IOI of the adjacent notes. When a performance rule is matched in the data in the score file, jPop-E multiplies the expansion (reduction) ratio prescribed in the rule sequentially with the initial (nominal) loudness and duration of each note and IOI of the adjacent notes.

Figure 4 shows the GUI for adjusting the control parameters. The user can modify these parameters by moving sliders.

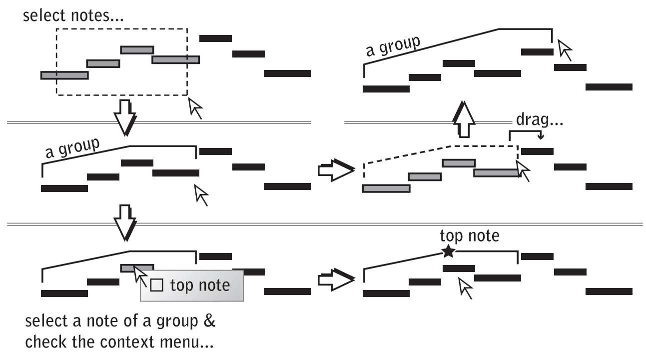

Figure 5: Editing a group and a top note

3.3 Editing Music Structures

(1) Group structure of parts. A group is a sequence of adjacent notes that constitutes a musical unit such as a phrase. A group is placed hierarchically according to each voice part. Voice parts are prohibited from having a nested group structure. jPop-E identifies each group by using the pair constituting the beginning and last notes of the group. We assume all the notes of a voice part constitute a group, and they belong to either of the adjacent groups. In adjacent groups, the beginning note of the preceding group and the last note of the following group can overlap. The starting time or ending time between two groups in different voice parts may not always coincide.

Figure 5 shows how to edit a group structure. To create a new group, the user selects the 'grouping' mode in the main menu, and then chooses the notes. When a user desires to change the bounds of a group, (s)he selects the corner of the group and expands/ reduces as desired.

(2) Top note of a group. The top note (tag) in a group is utilized for applying one of the rules of group expression (see section 2.1) as the transition of the dynamics and tempi of the phrase appear as a linear mountain. In Figure 5, to flag a note as the top note of a group, the user chooses the note and opens the context menu by clicking the right mouse button, then marks it with 'top note' from the menu.

(3) Attentive Part. The attentive part is a sequence of intermittently remarkable sounds in a piece that occur regardless of the role of the voice part. An example of an attentive part is shown in Figure 5. An attentive part is a sequence of notes that can be easily remembered and sung

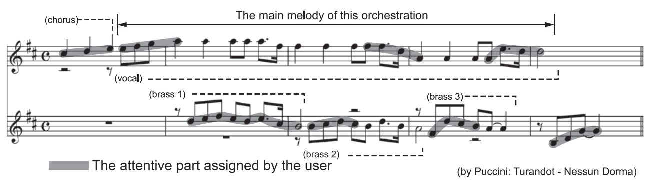

Figure 6: An example of the attentive part of a user.

to oneself. Although there is as yet no formal procedure to identify such parts, we note that such identification is natural and that identification of these parts would require little effort from users.

To set an attentive part, the user selects the 'attentive part' mode in the main menu, then chooses notes in the order of time, while referring to the registered group structure.

3.4 Managing Performance Rules

jPop-E provides the user with preset rules regulated by Pop-E (see 2.1) and enables the user to register new rules.

At present version of jPop-E provides a CUI (Character User Interface)- based function to register a rule which acts on each tag linked to each note. The user specifies the tag name and the target note (either the tag-given note or the preceding note), with the expansion/reduction ratio regarding the loudness and duration of each note and the IOI between the note and the next one. The newly registered rules are processed in the same way as the preset rules.

4. DISCUSSION

In this section, we discuss efforts to improve efficiency in performance rendering design.

jPop-E users' task involves iteration of listening to and evaluating a generated performance, adjusting the control parameters, and modifying the musical phrase structures. jPop-E provides preset performance expression rules and a default set of control parameters. The system enables a beginner to generate expressive performances without additional operations, if phrase structure expression is not necessary. If music group expressions are required, the user only has to enter a phrase structure by using the provided GUI. To modify a phrase structure, the user of another system might need special knowledge about music analysis. Hamanaka et al. developed an automatic generator system of phrase structure based on GTTM [7, 8]. They introduced adjustable parameters in order to provide GTTM users with a function for giving priority to rules; however, their system does not provide a function to generate a performance based on any expression model. Hirata et al.'s music summarization system provides a GUI editor for annotating phrase structures based on GTTM's time-span reduction [9]. However, the phrase structure of time-span reduction might not always match the structure of the performance expression.

Bresin et al. discussed the propriety of performance rules on the emotional expression of articulation (e.g., legato, staccato) on each note by comparing a rule-based system with a neural network system [10]. Then they developed pDM to control performance expression in the emotional space of happiness, sadness, and so on [11]. Their approach is an effective way to reduce user manipulations. However, we are afraid that the expression of adjectives would not always match the performance that the user imagines. Our system is able to control the performance expression directly.

5. SUMMARY

This paper gave an overview of a performance rendering model Pop-E (Poly-Phrase Ensemble) and described a newly developed Java-based performance design interface jPop-E.

By using jPop-E, MIDI data including expressive tempo changes or velocity control can be efficiently created based on the user's musical intention.

In the future, we would like to provide GUI functions to register new performance rules and parameter settings for them, and to test jPop-E with many users and gather feedback. We are also planning to analyze and consider reuse of data on phrase structures and parameters elaborated by test users. We would then improve jPop-E to make it more suitable for content design and music education.

6. REFERENCES

[1] L. Frydén and J. Sundberg, "Performance rules for melodies. origin, functions, purposes," in *Proc. of Intl. Computer Music Conf. (ICMC)*, pp. 221–224, 1984.

[2] G. Widmer and A. Tobudic, "Playing mozart by analogy: Learning multi-level timing and dynamics strategies," *Journal of New Music Research*, vol. 32, no. 3, pp. 259–268, 2003.

[3] R. Hiraga, M. Hashida, K. Hirata, H. Katayose, and K. Noike, "Rencon: toward a new evaluation method for performance," in *Proc. of Intl. Computer Music Conf. (ICMC)*, pp. 357–360, 2002.

[4] C. Raphael, "Orchestra in a box: A system for real-time musical accompaniment," in *Working Notes of IJCAI-03 Rencon Workshop*, 2003.

[5] H. Katayose and K. Okudaira, "sfp/punin: A performance rendering interface using expression model," in *Working Notes of IJCAI-03 Rencon Workshop*, 2003.

[6] M. Hashida, N. Nagata, and H. Katayose, "Pop-E: A performance rendering system for the ensemble music that considered group expression," in *Proc. of Intl. Conf. on Music Perception and Cognition (ICMPC)*, pp. 526–534, 2006.

[7] F. Lerdahl and R. Jackendoff, *A Generative Theory of Tonal Music*. MIT Press, 1983.

[8] M. Hamanaka, K. Hirata, and S. Tojo, "Automatic generation of grouping structure based on the gttm," in *Proc. of International Computer Music Conference (ICMC)*, pp. 141–144, November 2004.

[9] K. Hirata and S. Matsuda, "Interactive music summarization based on generative theory of tonal music," *Journal of New Music Research*, vol. 32, no. 2, pp. 165–177, 2003.

[10] R. Bresin and G. Battel, "Articulation strategies in expressive piano performance. analysis of legato, staccato, and repeated notes in performances of the andante movement of mozart's sonata in g major (k. 545)," *Journal of New Music Research*, vol. 29, no. 3, pp. 211–224, 2000.

[11] A. Friberg, "pdm: an expressive sequencer with real-time control of the kth music performance rules movements," *Computer Music Journal*, vol. 30, no. 1, pp. 37–48, 2006.