WEB-BASED TIME-SPAN TREE EDITOR AND ANALYSIS DATABASE

Masatoshi Hamanaka
RIKEN, Tokyo
masatoshi.hamanaka@riken.jp

Yui Isono
RIKEN, Tokyo
yui.isono@riken.jp

Keiji Hirata
Future University Hakodate, Hakodate
hirata@fun.ac.jp

Satoshi Tojo
JAIST, Nomi
tojo@jaist.ac.jp

ABSTRACT

This paper presents a Web-based manual time-span tree generation tool. While we have previously collected time-span trees analyzed by musicologists on the basis of the generative theory of tonal music and created several editing tools for the trees, the tools are not ideal for practical use due to the following four issues. First, the piano roll dis-play tool does not allow musicologists to conduct intuitive analysis. Second, the staff score display library can no longer be used when editing time-span trees created on an iPad due to an OS update. Third, with all editing tools, operation slows when the number of notes increases. Fourth, editing a tree-structured branch requires a large number of manual operation steps. In response to these issues, we developed a Web-based manual time-span tree editor that displays staff scores and enables high-speed operation by reviewing the data structure. Experimental results indicate that the number of operation steps decreased by about 14.6%.

1. INTRODUCTION

Our objective is to develop a music operation system that uses time-span trees constructed on the basis of generative theory of tonal music (GTTM) analysis [1-3]. Music production is complex because it requires a large number of operation steps [4]. For example, making a melody simpler requires many operation steps. These operation steps can be performed using a time-span tree for performance rendering, music reproduction, and summarization of the music [5]. The goal of our system will be to have a time-span tree that reduces the number of operation steps.

The main advantage of using time-span tree for music production is using it reduce the number of notes in a melody and extract a simple melody reduced from a complex melody [6, 7]. With this reduction, the relationship between the complex melody and reduced melody is expressed as a subsumption relationship. There is also a "meet" operation that extracts the common part of two time-span trees and a "join" operation that overlaps two trees [8]. The melody-morphing method is a combination of the subsumption relation described above and the meet and join operations [9]. *ShakeGuitar* [10] and *Melody Slot Machine* [11] are applications that use the melody-morphing method.

Figure 1 shows an example of how to reduce the number of notes in a melody using a time-span tree of K. 331/300i (Mozart's Piano Sonata No. 11). We can obtain an abstracted melody by slicing the tree in the middle and omitting notes whose branch connections are below the line. Therefore, time-span reduction is the inverse process of composition.

The largest problem in constructing a melody operation system using a time-span tree is the low accuracy of the time-span tree analyzer [12, 13]. We previously carried out a naive implementation of GTTM, but the analysis accuracy of the time-span tree was just 60% [12]. The analysis accuracy of the time-span tree of a probability-based analysis system that introduces probabilistic context-free grammar is 76% [13], which is a little better, but since that evaluation is limited to the melody for which the probability has been obtained, further improvement is required.

In GTTM analysis, a grouping structure, metrical structure, and time-span tree are extracted in that order. We recently implemented deep learning to extract grouping and metrical structures, and the analysis performance improved significantly [14]. Extracting a time-span tree by deep learning was not possible due to the lack of analyzed data.

Therefore, in this work we aimed to add analysis data, but this is difficult to do with conventional analysis tools. For example, while the *interactive GTTM analyzer* [15] can find simple analysis mistakes, it is not suitable for musician analysis because the display is a piano roll. While the *manual time-span generation tool* [16] is able to display staff scores, an OS update made the staff score display library unusable. Other issues include the lengthy calculation times and high number of operation steps.

To resolve these issues, we developed an analyzer that uses the popular Web-based staff score display library. We also improved the data structure and manipulation methods. We compared the number of operation steps with this Web-based time-span tree editor with those of the manual time-span generation tool. Experimental results indicated that the number of operation steps decreased by about 14.6%.

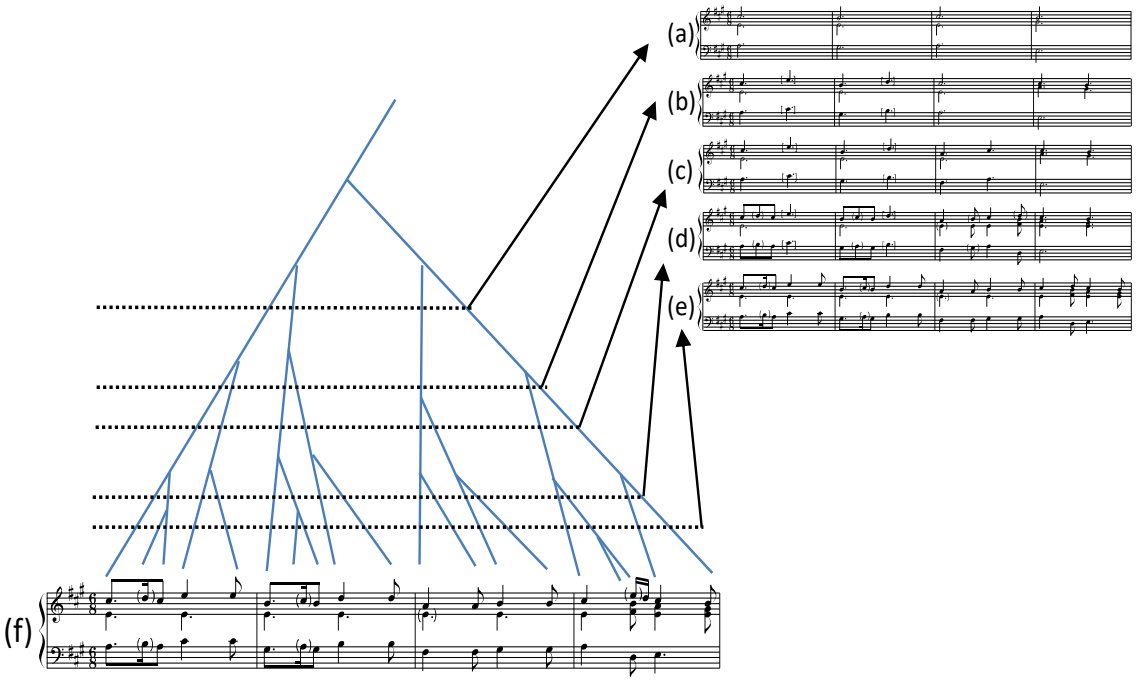


Figure 1. Reduction of melody.

In Section 2 of this paper, we discuss the issues with previous time-span tree editors. In Section 3, we present our Web-based time-space tree editor and experimental results in Sections 4 We conclude in Section 5 with a brief summary and mention of future work.

2. ISSUES WITH PREVIOUS TIME-SPAN TREE TOOLS

We have been building GTTM analyzers and time-span tree editors for over 15 years, but the main ones are the interactive GTTM analyzer and manual time-span generation tool.

2.1 Interactive GTTM analyzer

One of the interactive GTTM analyzers we developed is shown in Fig. 2. Because musicians are accustomed to staff scores, it is difficult for them to analyze data presented as a piano roll. For example, in a piano roll, as shown in Fig. 3(a), it may be difficult to distinguish between a series of notes of the same pitch or a single long note. When the number of notes in a piece increases, the number of branches in the time-span tree also increases. Therefore, it is difficult to visually distinguish between an essential parent branch and an ornamented child branch in a time-span tree because the parent branches and their child branches are mostly straight lines (Fig. 3(b)).

Figure 4 shows the position (highlighted) where the selected branch can be connected in the time-span tree. Thus, if we move the branch to the target position, that branch becomes highlighted then connected to the desired position. The largest problem with this tool is that many actions are required for the simplest and most frequently performed operation steps, such as swapping the parent and child of a branch.

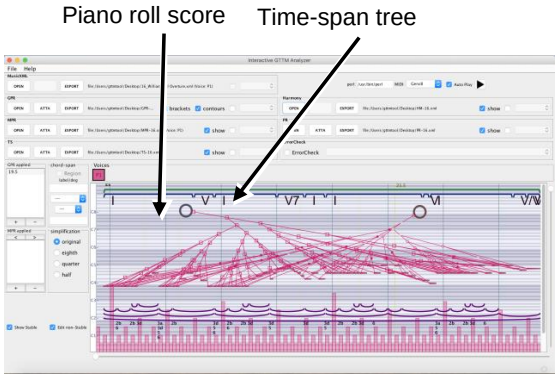
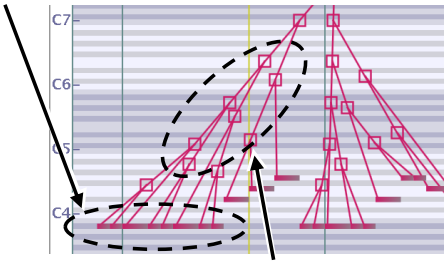


Figure 2. Interactive GTTM analyzer with piano roll.

(a) Series of notes



(b) Parent and child branches

Figure 3. Difficulty in manipulating musical piece with large number of bars and notes.

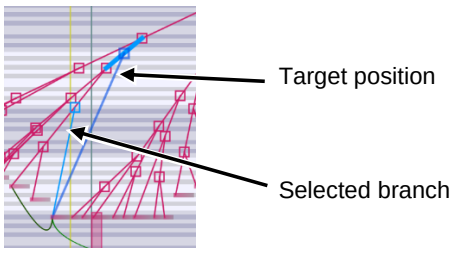


Figure 4. Highlighting the position that can connect the selected branch in previous editor for editing time-span tree.

2.2 Manual time-span tree generation tool

The manual time-span tree generation tool can display staff scores and zoom in on the screen, making branch operations easier than with the interactive GTTM analyzer. However, it comes with its own problems.

Since it is difficult to touch and select single notes or branches with one's fingertip, we added circular handles to them. These handles are attached to all notes except rests and grace notes. After loading the Music XML, the handles are visible directly above the notes (Fig. 5(a)) [16]. The heights of the handles are automatically separated because if two handles are very close to each other, it may not be possible to select the desired one. By simply placing one handle over another one, we can generate a tree structure such that the original handle becomes a sub-branch and the handle that was placed on top becomes the parent branch (main branch) (Figs. 5(b) and (c)).

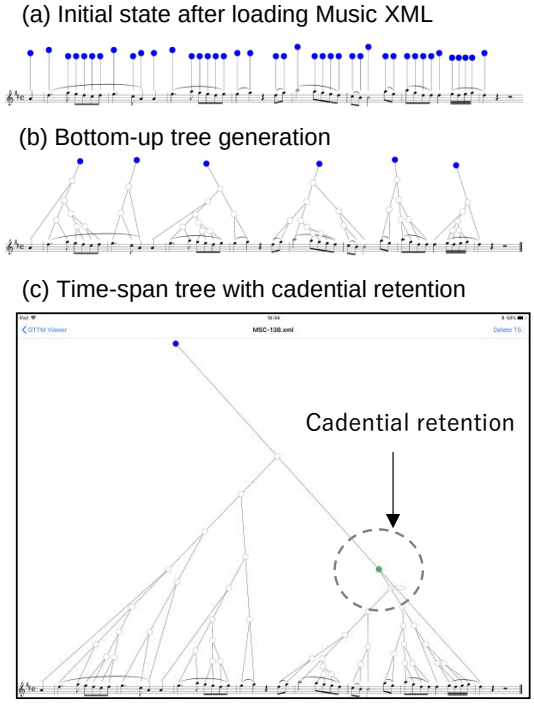


Figure 5. Screenshot of manual time-span tree generation tool.

By simply placing a handle on another handle, a tree structure can be generated in a way that the original handle becomes a sub-branch and the handle that was placed on top becomes the parent branch (main branch) (Figs. 7(b) and (c)).

This tool has no problem assembling a time-span tree from the bottom up, as described above. However, modifying a tree once it has been completed cannot be done in a single action. For example, to disconnect a branch, one must first press and hold the handle to bring up a menu then select which connection to delete from the menu items.

Furthermore, since only the handle can be connected, if one wants to modify the tip of a tree branch, one needs to cut the tree structure up to that point and recreate it again from the bottom up.

2.3 Issue with data structure

Time-span trees are stored by time-span XML [17, 18]. Figure 6(a) shows a time-span XML corresponding to the time-span tree in (b). The time-span XML consists of *ts*, primary, and secondary elements. The *ts* element indicates the length and position of the time span in a piece of music. It includes a head element, which requires a note element indicating the most salient note in the time span. If there is more than one note in the time span, we can divide the time span into two parts: one that includes the head and one that does not. The primary element in the *ts* element has a next-level ts element that corresponds to the time span, which includes the upper-level head. The secondary element in the *ts* element has a next-level *ts* element that corresponds to the time span, which does not include the upper-level head.

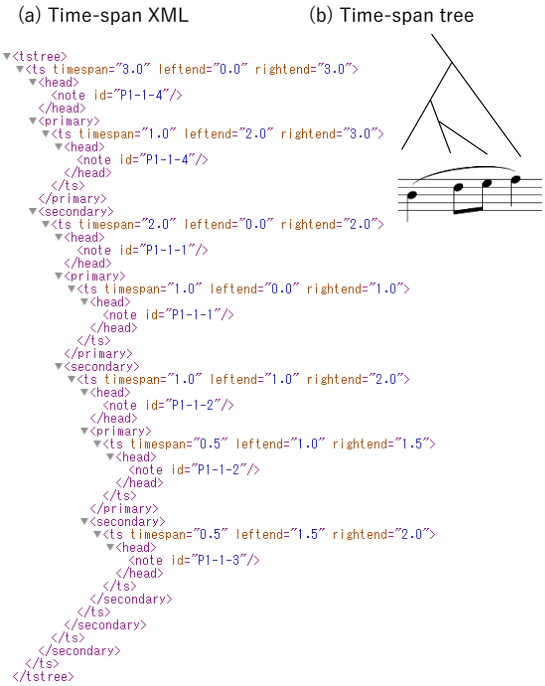


Figure 6. Time-span XML and time-span tree.

This storage format is suitable for operation steps such as melody reduction, since the *ts* element indicates the length of the time span, that is, the note value after reduction.

However, it is not suitable for the data format when displaying the time-span tree. The time-span tree is displayed with the branches equally spaced to make it easy to see even when the number of branches increases. For example, when there are five child branches in the parent branch, the intervals between the five branch points are the same, i.e., equally spaced. To do this, the display program needs to know how many branches come out of one parent branch. In time-span XML, this information is not explicitly given, so it is necessary to recursively search the file.

3. WEB-BASED TIME-SPAN TREE EDITOR

In response to the issues discussed in the previous section, we developed our Web-based time-span tree editor, which is an improvement upon the manual time-span tree generation tool. The time-span tree is created from the bottom up, the same as the manual time-span tree generation tool, but it can be easily modified. The four features of the Web-based time-span tree editor are described in the following subsections.

3.1 Single action for connecting and removing

The Web-based time-span tree editor reduces the number of operation steps by connecting and deleting branches in a single action. That is, a connection is established simply by selecting a branch and attaching it to another branch, and a connection is deleted simply by selecting and moving the connected branch (Fig. 7). With the manual time-span tree generation tool, branch connection is a single action but deletion requires a double action. This is because there are two operation steps for the connected branch—deletion of the branch and designation of credential retention—and it is necessary to pull up a menu and select one (Fig. 5(c)). Since the cadence retention is specified only once per song, it is very inefficient to use a double action for deletion. With the Web-based time-span tree editor, the credential retention is specified by tapping the "c" button on the keyboard, allowing a single action for deletion. When the branches are connected, the positions of the branches are recalculated and automatically displayed again so that the connection points of the branches are equally spaced.

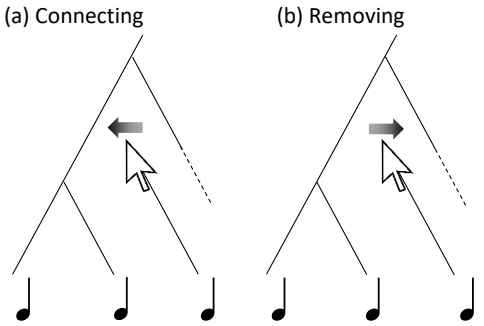


Figure 7. Connecting and removing a branch.

3.2 Selecting candidate time-span trees

If the branches are dense, they may be connected to places different than the user intends. If the connection is made in an unintended place, additional work is required because the branch must be removed then connected again. Moreover, the reconnection is not always successful.

With the Web-based time-span tree editor, users can select the time-span tree candidates that they intend. Since the branches of the time-span tree do not intersect, the place where the branch of a note n5 is attached can be any of P_1, P_2, P_4, and P_5, as shown in Fig. 8(a). Figure 8(b) shows a case in which the connection to P_3 is made by a user operation. In this case, the user may have actually wanted P_2 or P_4, which are adjacent to P_3. After connecting to P_3, the user can simply tap the space bar to select (c) P_2 or (d) P_4 instead.

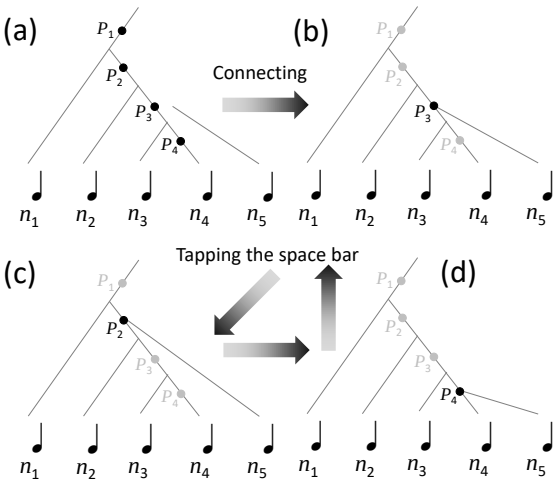


Figure 8. Selecting candidate time-span trees.

3.3 Defining data structure using JSON

We use JavaScript Operation Notation (JSON) to define data structures that are suitable for display in computer programs and easy for users to understand [19].

A time-span tree and its subtrees are surrounded by curly brackets. A subtree may be a single branch, include other branches, or include subtrees. The subtree of the time-span tree being edited is described in time series order, with a colon and square brackets indicating descriptions after the branch. Child branches of the branch are described with square brackets. If there are multiple child branches, they are described in comma-separated order near the root. Figure 9 shows an example of time-span JSON, where n1 to n4 contain the note IDs of the corresponding Music XML.

3.4 Web-based implementation with music notation rendering API and analyzed database

We implemented our tool using HTML5 (Fig. 10), which is advantageous because it can operate on multiple platforms (Windows, Mac, Linux, etc.) [20]. Java's interactive GTTM analyzer also runs on multiple platforms, but it has

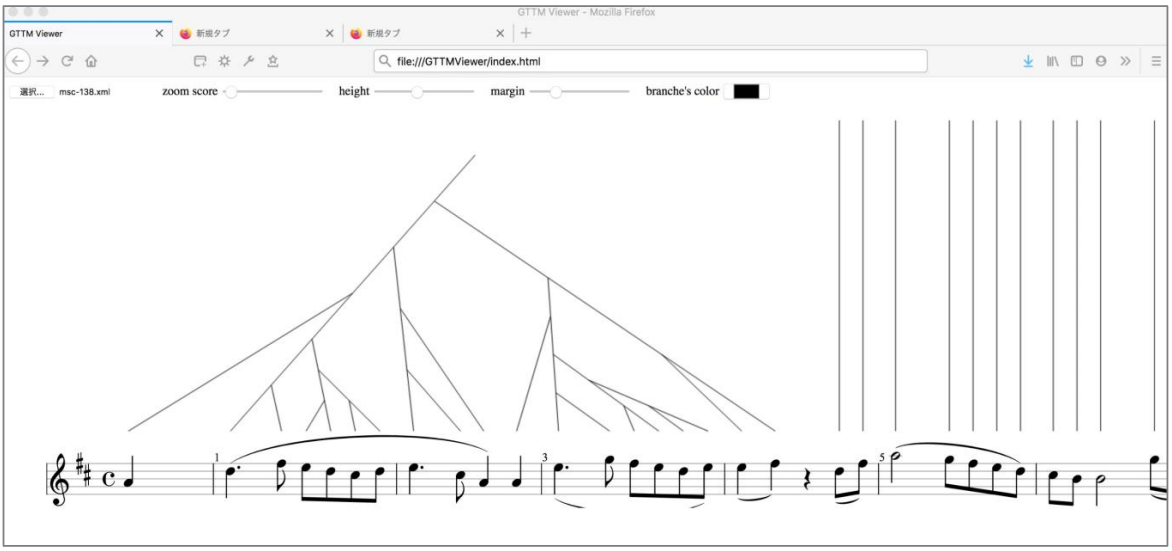


Figure 9. Web-based time-span tree editor.

limitations such as some instructions working only on Mac. We used the OpenSheetMusicDisplay music notation rendering API to display staff scores [21]. OpenSheetMusicDisplay is made with JAVA script and can read Music XML and display the staff score on the browser.

(a) Time-spanJSON

```
[
  {"mom": "P1-1-1", "children": []},
  {"mom": "P1-1-3", "children": [
      {"mom": "P1-1-4", "children": []},
      {"mom": "P1-1-2", "children": []}
    ]
  }
]
```

(b) Time-span Tree

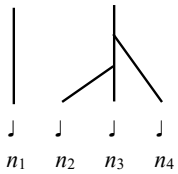


Figure 10. (a) Time-span JSON and (b) time-span tree.

4. EXPERIMENT

We compared the number of operation steps with the Web-based time-span tree editor with those with the manual time-span generation tool, as the manual time-span generation tool no longer runs.

If we create a time-span tree from the bottom up, the number of operation steps is the same for the Web-based time-span tree editor and manual time-span generation tool. The number of steps mentioned here is the number of operation steps performed with a single mouse click. In practice, branches are often mistakenly attached, and the number of steps increases due to correcting such errors. Since the manual time-span generation tool can no longer run, we compared the number of operation steps as follows.

First, the participants created a time-span tree using the Web-based time-span tree editor and captured the display screen with a moving image. Next, we counted the number of operation steps of the editor while watching the moving image. The same operation was converted into the number of operation steps when the operation was performed using the manual time-span generation tool. The results (Table 1) indicate that the number of operation steps decreased by about 14.6%. The five pieces in Table 1 were selected randomly from the GTTM database [17].

Title of piece	Operation steps of Web-based editor	Operation steps of manual generation tool
1. Carmen	40	48
2. Greensleeves	20	22
3. Amazing Grace	25	28
4. Corrente	37	45
5. Vittoria, mio core!	24	28
Avg. of five pieces	29.2	34.2

Table 1. Average number of operation steps for constructing time-span tree.

5. CONCLUSION

We developed the Web-based time-span tree editor. Our main contributions are as follows.

- **Development of a Web-based time-span tree editor using staff score**
  The conventional analyzer using a piano roll score and an analyzer on an iPad are OS-dependent programs that have been difficult to use. We expect that our Web-based editor can be used by many people as it is a multi-platform Web-based application.
- **Time-span tree editing with a single action for connecting and removing branches**
  We simplified the operation method and reduced the number of operation steps. We plan to enable further reduction by eliminating time-span trees that violate time-span tree well-formedness rules.

Our Web-based time-span tree editor and analysis database can be accessed at `http://gttm.jp/gttm/`. Our future work will involve adding analysis data by using the Web-based time-span tree editor.

Acknowledgments

This work was supported by JSPS KAKENHI Grant numbers 17H01847 and 16H01744.

6. REFERENCES

[1] F. Lerdahl and R. Jackendoff, *A Generative Theory of Tonal Music*, *MIT Press*, 1985.

[2] F. Lerdahl: *Tonal Pitch Space*, Oxford University Press, 2001.

[3] F. Lerdahl: *Composition and Cognition: Reflections on Contemporary Music and the Musical Mind*, University of California Press, 2019.

[4] R. U. Nelson: *The Technique of Variation; A Study of the Instrumental Variation from Antonio de Cabezón to Max Reger*, University of California Press, 1948.

[5] K. Hirata and S. Matsuda, "Interactive music summarization based on generative theory of tonal music," *Journal of New Music Research*, pp. 165–177, 2003.

[6] N. C., Hansen, "The Legacy of Lerdahl & Jackendoff's A Generative Theory of Tonal Music: Bridging a significant event in the history of music theory and recent developments in cognitive music research," *Danish Yearbook of Musicology*, 38, pp. 33–55, 2001.

[7] C. Tsougras, "Analysis of Early 20th century Chromatic Modal Music with the use of the Generative Theory of Tonal Music - Pitch Space and Prolongational issues in selected Modal Idioms," in *Proceedings of the 7th Triennial Conference of European Society for the Cognitive Sciences of Music* (*ESCOM 2009*), 2009, pp.531–539.

[8] K. Hirata and T. Aoyagi, "Computational Music Representation Based on the Generative Theory of Tonal Music and the Deductive Object-Oriented Database," *Computer Music Journal*, 27:3, pp. 73–89, 2003.

[9] M. Hamanaka, K. Hirata, and S. Tojo, "Melody Morphing Method Based on GTTM," in *Proceedings of International Computer Music Conference* (*ICMC2008*), 2008, pp. 155–158.

[10] M. Hamanaka, M. Yoshiya, and S. Yoshida, "Constructing Music Applications for Smartphones," in *Proceedings of the 2011 International Computer Music Conference* (*ICMC2011*), 2001, pp. 308–311.

[11] M. Hamanaka, "Melody Slot Machine: A Controllable Holographic Virtual Performer", in *Proceedings of the 27th ACM International Conference on Multimedia* (*MM'19*), 2019, pp. 2468–2477.

[12] M. Hamanaka, K. Hirata, and S. Tojo, "Implementing a generative theory of tonal music," *Journal of New Music Research*, pp. 249–277, 2006.

[13] M. Hamanaka, K. Hirata, and S. Tojo, "σGTTM III: Learning-Based Time-Span Tree Generator Based on PCFG," in *Proceedings of International Symposium on Computer Music Multidisciplinary Research* (*CMMR2015*), 2015, pp. 387–404.

[14] M. Hamanaka, K. Hirata, and S. Tojo, "deepGTTM-III: Simultaneous Learning of Grouping and Metrical Structures," in *Proceedings of International Symposium on Computer Music Multidisciplinary Research* (*CMMR2017*), 2017, pp. 161–172.

[15] M. Hamanaka and S. Tojo, "Interactive GTTM Analyzer," in *Proceedings of International Conference on Music Information Retrieval Conference* (*ISMIR2009*), 2009, pp. 291–296.

[16] Makemusic Inc., "MusicXML DTD," Available online at: https://www.musicxml.com/for-developers /musicxml-dtd/, Retrieved May, 2020.

[17] M. Hamanaka, K. Hirata, and S. Tojo: "GTTM Database and Manual Time-span Tree Generation Tool," in *Proceedings of the 15th Sound and Music Computing Conference* (*SMC2018*), 2018, pp. 462–467.

[18] W3C: "Extensible Markup Language (XML)," Available online at: https://www.w3.org/XML/, Retrieved May, 2020.

[19] ECMA International: "The JSON Data Interchange Syntax," 2nd Edition, ECMA-404, Available online at: http://www.ecma-international.org/publications/ files/ECMA-ST/ECMA-404.pdf, Retrieved May, 2020.

[20] W3C: "HTML5.2 W3C Recommendation," Available online at: https://www.w3.org/TR/html52/, Retrieved May, 2020.

[21] Phonicscore Inc., "OpenSheetMusicDisplay," Available online at: https://opensheetmusicdisplay. org/, Retrieved May, 2020.