INTERACTIVE GTTM ANALYZER

Masatoshi Hamanaka
University of Tsukuba

hamanaka@iit.tsukuba.ac.jp

Satoshi Tojo
Japan Advanced Institute of
Science and Technology
tojo@jaist.ac.jp

ABSTRACT

We describe an interactive analyzer for the generative theory of tonal music (GTTM). Generally, a piece of music has more than one interpretation, and dealing with such ambiguity is one of the major problems when constructing a music analysis system. To solve this problem, we propose an interactive GTTM analyzer, called an automatic time-span tree analyzer (ATTA), with a GTTM manual editor. The ATTA has adjustable parameters that enable the analyzer to generate multiple analysis results. As the ATTA cannot output all the analysis results that correspond to all the interpretations of a piece of music, we designed a GTTM manual editor, which generates all the analysis results. Experimental results showed that our interactive GTTM analyzer outperformed the GTTM manual editor without an ATTA. Since we hope to contribute to the research of music analysis, we publicize our interactive GTTM analyzer and a dataset of three hundred pairs of a score and analysis results by musicologist on our website http://music.iit.tsukuba.ac.jp/hamanaka/gttm.htm, which is the largest database of analyzed results from the GTTM to date.

1. INTRODUCTION

We have been constructing a music analyzer based on the generative theory of tonal music (GTTM) [1]. The GTTM is composed of four modules, each of which assigns a separate structural description to a listener's understanding of a piece of music. These four modules output a grouping structure, a metrical structure, a time-span tree, and a prolongational tree. The main advantage of implementing the GTTM on a computer is to acquire tree structures called time-span and prolongational trees from the surface structure of a piece of music. The time-span and prolongational trees provide melody morphing, which generates an intermediate melody between two melodies with a systematic order [2]. It can also be used for performance rendering [3-5] and reproducing music [6] and provides a summarization of the music. This summarization can be used as a representa-

tion of a search, resulting in music retrieval systems [7].

In computer implementation of music theory [1, 8-10], we have to consider two types of ambiguity in music analysis. One involves human understanding of music, and the other concerns the representation of music theory. The former tolerates our subjective interpretation, while the latter is caused by the incompleteness of formal theory, and the GTTM is not an exception. Therefore, due to the former's ambiguity, we assume there is more than one correct result.

In our previous work, we proposed the exGTTM (machine-executable extension of GTTM) and constructed an automatic time-span tree analyzer (ATTA) to avoid the latter type of ambiguity, introducing as many parameters as possible [11, 12]. Whenever we find a correct result that the exGTTM cannot generate, we add new parameters with proper values to improve the result.

However, the ATTA has been clumsy for the first type of ambiguity. Even an identical melody can be played in different ways to represent different feelings since the ATTA cannot output the different analysis results in the same melody repetition. To solve this problem, we developed a GTTM manual editor that manually alternates the analysis results of the ATTA, according to the user's interpretations of a piece of music.

However, the ATTA still exhibits problems concerning the latter type of ambiguity. For example, the GTTM consists of feed-back operations from higher- to lower-level in the tree structure; however, no detailed description and only a few examples are given. To solve this problem, we developed a GTTM process editor, which enables seamless change of the automatic analysis process with an ATTA and the manual edit process with a GTTM manual editor. Therefore, a user can acquire the target analysis results by iterating the automatic and manual processes interactively and easily reflect his or her interpretations on a piece of music.

This paper is organized as follows. We present an overview of our interactive GTTM analyzer, which consists of the ATTA, GTTM manual editor, and GTTM process editor in Section 2, propose a manual editing method of the GTTM manual editor in Section 3, propose a process editing method of the GTTM process editor in Section 4, and present experimental results and conclusions in Sections 5 and 6, respectively. Finally, we provide in the appendix the data format of the analyzing results of the GTTM, which we publicize along with those of the interactive GTTM analyzer.

2. INTERACTIVE GTTM ANALYZER

Figure 1 is a screenshot of the viewer of our interactive GTTM analyzer. There is a sequence of notes displayed in a piano roll format. Below the notes is a grouping structure, which is graphically presented as several levels of arcs. The grouping structure is intended to formalize the intuitive belief that tonal music is organized into groups that are in turn composed of subgroups. Below the grouping structure is a metrical structure. The metrical structure describes the rhythmical hierarchy of the piece by identifying the position of strong beats at the levels of a quarter note, half note, one measure, two measures, four measures, and so on. Strong beats are illustrated as several levels of bars. Above the notes, there is a time-span tree. The time-span tree is a binary tree, which is a hierarchical structure describing the relative structural importance of notes that differentiate the essential parts of the melody from the ornamentation. Below the time-span tree is a prolongational tree, a binary tree that expresses the structure of tension and relaxation in a piece of music.
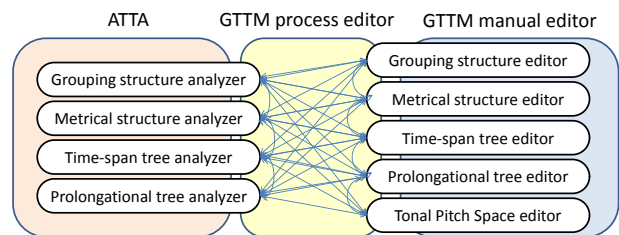
Figure 2 is an overview of our interactive GTTM analyzer, consisting of an ATTA, a GTTM manual editor, and a GTTM process editor. The ATTA consists of a grouping structure, a metrical structure, and time-span tree analyzers. We have been developing a prolongational tree analyzer. Hamanaka et al. explain the details of the ATTA [11].

The GTTM manual editor consists of grouping, metrical, time-span, prolongational, and Tonal Pitch Space editors. The Tonal Pitch Space [12] is a music theory for chord progression composed by Lerdhal, who is one of the authors of the GTTM. Although the GTTM includes rules that require the analysis results of chord progression, the ATTA uses such rules by adopting the results of the Tonal Pitch Space.

The analyzing process with the ATTA and GTTM manual editor is complicated, and sometimes a user is confused as to what he or she should do in the next process, as there are three analyzing processes in the ATTA and five editing processes in the GTTM manual editor. A user may iterate the ATTA and manual edit processes multiple times. To solve this problem, we propose a GTTM process editor, which presents candidates for the next process of analysis, and a user only needs to change the process, just by selecting the next process.

We use an XML format for all the input and output data structures of our interactive GTTM analyzer. Each analyzer and editor of our analyzer works independently, but

Figure 2. Overview of interactive GTTM analyzer.

they are integrated with the XML-based data structure.

3. GTTM MANUAL EDITOR

In some cases, the ATTA may produce a preferable result, which reflects the user's interpretation, but in other case it may not. When a user wants to change the analysis result according to his or her interpretation, he or she can use the GTTM manual editor. We describe the method for editing and constructing a musical structure of the GTTM using the GTTM manual editor.
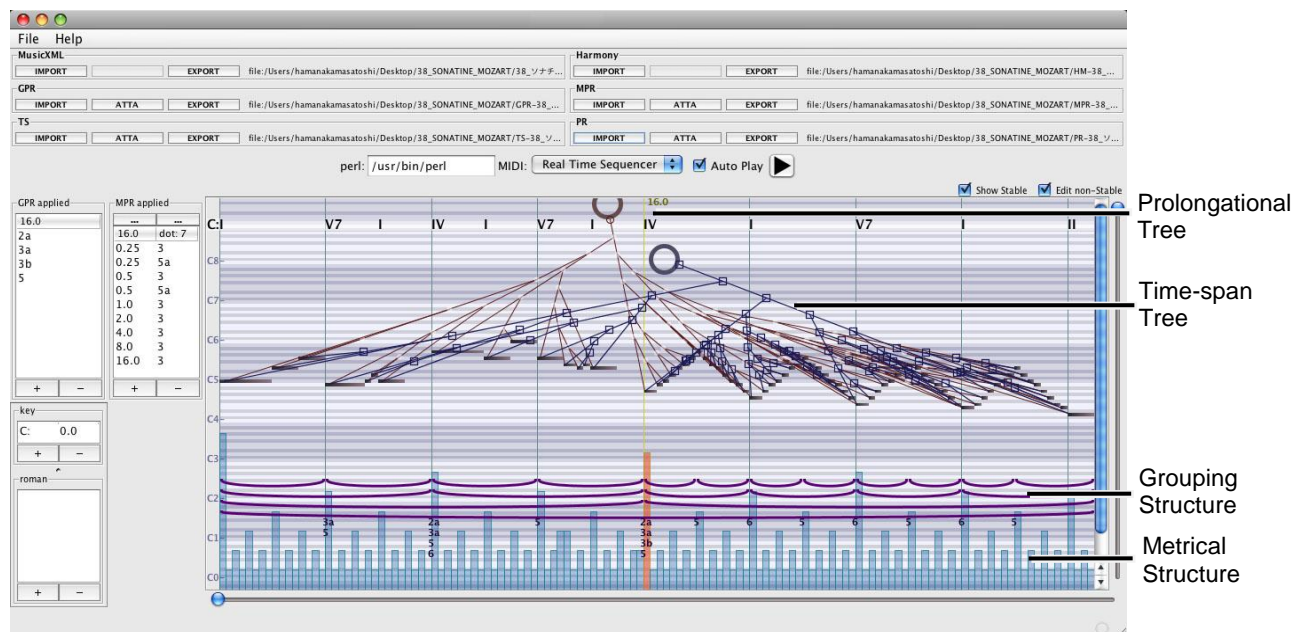
Figure 1. Screenshot of interactive GTTM analyzer.

3.1 Grouping structure editor

Figure 3 is a screenshot of our interactive GTTM analyzer in editing a grouping structure. The color of the target group and all its subgroups turn red after selection with a mouse. Then we can open a popup menu by right clicking the mouse. There are four operations in the popup menu, *divide this group and create subgroup*, *divide this group*, *delete*, and *delete descendant*.

To change a position of a grouping boundary, a user first delete the groups which adjoin the boundary then divide the upper level (global level) group and create new subgroups where he or she wants to create a boundary. By left clicking a grouping boundary, the user sees the rules that are applied to the boundary and he or she can add or delete these rules.

Figure 3. Screenshot of grouping structure editor.

3.2 Metrical structure editor

Although the metrical structure analyzer in the ATTA performs fairly well [11], a user may want to slightly edit the metrical structure. In which case, he or she applies the metrical structure editor, and changes the strength level of a beat by dragging a bar up or down. At the same time, he or she sees the rules that are applied to the bar and can add or delete these rules.

While editing beat strength, a user may break hierarchical metrical structures. In other words, the results of the metrical structure editor sometimes do not hold for the metrical preference rules. This problem can be solved using the GTTM process editor, which we discuss in Section 4.

3.3 Time-span tree editor

In the time-span tree, each branch has a head represented by a square in the time-span tree editor, and a user can move the head by dragging another branch. Figure 4 is a screenshot of dragging a head. The light blue branch is the former position, and the dark blue branch is the latter position. A user can select a type for each head by opening the popup menu among those four types, *ordinary*, *fusion*, *transformation*, and *cadential retention*.

3.4 Prolongational tree editor

The process of the prolongational tree editor is the same as that for the time-span tree. The prolongational tree is constructed by reconnecting the heads based on the time-span tree. There are head connection constraints of the
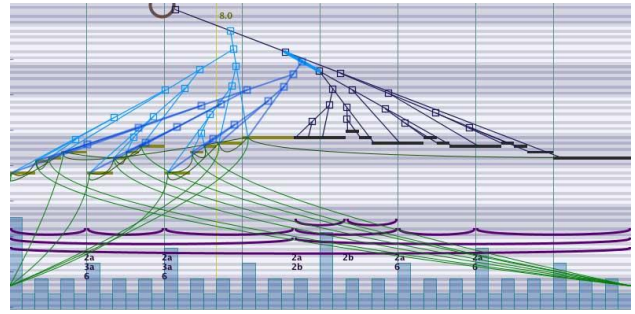

Figure 4. Screenshot of when dragging head.

prolongational tree. When a head connection of a prolongational tree is ill-formed, the GTTM process editor automatically opens the popup menu and displays candidates for a solution.

3.5 Tonal Pitch Space editor

The reason we include a Tonal Pitch Space editor in our interactive GTTM analyzer is that the editor provides quantitative grounds for the prolongational tree to be hierarchical. Therefore, analyzing the Tonal Pitch Space with the prolongational tree improves analyzing performance.

4. GTTM PROCESS EDITOR

There are two types of rules in the GTTM, which are *well-formedness* and *preference*. Well-formedness rules are necessary conditions for the rules assignment of a structure as well as the restrictions on the structure. When more than one structure satisfies the well-formedness rules, the preference rules indicate the superiority of one structure over another.

In the GTTM, the analysis sequence proceeds from the grouping structure, secondly to the metrical structure, next to the time-span tree, and finally to the prolongational tree. However, the GTTM contains feedback links from higher- to lower-level structures. For example, grouping preference rule 7 (GPR7) (time-span and prolongational stability) prefers a grouping structure that results in a more stable time-span and/or prolongation reduction. Therefore, to analyze with feedback link rules, we need to perform several analyzing processes by trial and error. The GTTM process editor helps in this repetition by performing three functions, data inputting, history recording, and process controlling.

4.1 Data inputting

Data inputting helps with the input of analysis results, which are prepared by another user or analyzer. For example, we do not have an automatic analyzer for the Tonal Pitch Space in our interactive GTTM analyzer; however, attempts have been made to implement the Tonal Pitch Space, so we can use those results [13].

We can add new rules to the ATTA using data inputting. For example, grouping preference rule 6 (GPR6) is a rule for parallelism in a grouping structure; however, the GTTM does not define the decision criteria for construing

whether two or more segments are parallel or not. Therefore, many implementations of GPR6 would be possible, although we propose only one of them. By adding a new rule to the ATTA, we can control a new adjustable parameter for the new rule, GPR6+, which is the new implementation of GPR6.

4.2 History recording

History recording records the operation of analysis, and a user can return to the previous phase of analysis. History recording enables the copying and pasting of several operations of analysis while editing parallel phrases.

In the GTTM, there are few descriptions of the reasoning and working algorithms needed to compute the analysis results, especially for the time-span and prolongational trees. By using history recording, we look forward to storing the analysis knowledge, which improves automatic analysis.

4.3 Process controlling

Process controlling enables seamless change of the analysis process by using the ATTA and the manual edit process by using the GTTM manual editor, representing candidates for the next process of analysis. The representation method differs depending on the number of candidates for the next process.

4.3.1 One candidate

When there is only one candidate process, the process-controlling function automatically executes the process. For example, when a user edits the strongest beat in Figure 5a in the 2nd level, the hierarchical metrical structure is broken because in level 3 of Figure 5b there are three weak continuous beats, and the metrical well-formedness rule 2 (MWFW2) does not hold. MWFR2 requires that strong beats are spaced either two of three beats apart at each metrical level. The process editor automatically alternately produces strong and weak beats in level 3 (Figure 5c). If there is a higher metrical structure than level 3, the metrical analyzer of the ATTA automatically analyzes after level 3 and constructs a hierarchical metrical structure reflecting the user's intention.
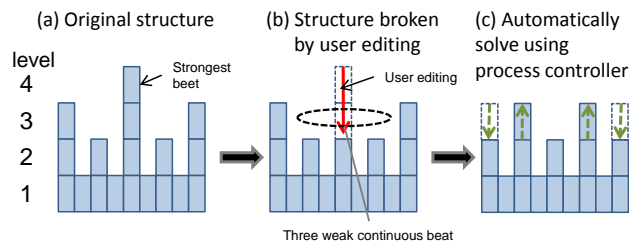
Figure 5. Automatically correct broken metrical structure.

4.3.2 A few candidates

When there are a few candidates, the process controlling function automatically opens the popup menu and shows the candidates. For example, if there is a grouping struc-

ture, as shown Figure 6a, and a user deletes a group at the upper left (Figure 6b), the grouping structure of Figure 6b is broken since grouping well-formedness rule 3 (GWFR3) does not hold. GWFR3 requires constraints that a group may contain smaller groups. To solve this problem, there are only two processes:

- Delete all the groups at the same level of the deleted group (Figure 6c).
- Extend the grouping boundary of the left end of the right group of the deleted group to the left end of that deleted group (Figure 6d).

The next process can be executed by selecting one of the two processes displayed in the popup menu.
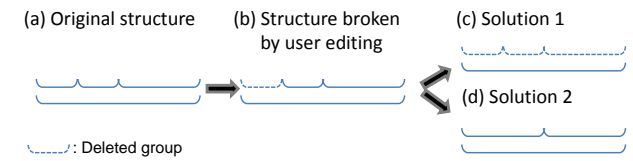
Figure 6. Two types of solutions for broken grouping structure.

4.3.3 Many candidates

When there are many candidates, the process-controlling function selects and shows the top-ten candidates from the history recording. The candidates are ordered depending on the similarity of the history. For example, after editing a time-span tree with the time-span tree editor, executing a grouping analyzer or metrical analyzer in the ATTA is ranked in the upper levels because there are rules for feedback link such as GPR7 or metrical preference rule 9 (MPR9). GPR7 (time-span and prolongational stability) is a link from the time-span and prolongational trees to the grouping structure, and MPR9 (time-span interaction) is a link from the time-span tree to the metrical structure.

We have not implemented the original ATTA on GPR7 and MPR9. In this paper, we omit the details of the implementation of these rules due to space limitations.

5. EXPERIMENTAL RESULTS

We asked a musicologist expert to manually analyze the score data faithfully with regard to the GTTM using our interactive GTTM analyzer. The musicologist collected three hundred 8-bar-long, monophonic, classical music pieces including notes, rests, slurs, accents, and articulations entered manually with music notation software called *Finale* [14]. The musicologist needed ten to twenty minutes for analyzing a piece. Three other experts crosschecked these results.

We measured the operating time for acquiring the target analysis results of our interactive GTTM analyzer and compared it with that of the GTTM manual editor without an ATTA. For the target analysis, we used one hundred pieces from the three hundred pairs of scores and correct data of grouping structure, metrical structure, and time-span tree. We did not use the prolongational tree in this

measurement since its analyzer is still under development. As a result, our interactive GTTM analyzer outperformed the GTTM manual editor without an ATTA (Table 1).

Melodies	Interactive GTTM analyzer	GTTM manual editor
1 Grande Valse Brillante	326 sec	624 sec
2. Moments Musicaux	541 sec	791 sec.
3. Turkish March	724 sec	1026 sec
4. Anitras Tanz	621 sec	915 sec.
5. Valse du Petit Chien	876sec.	1246 sec.
	:	:
Total (100 melodies)	575 sec.	891 sec.

Table 1. Operation time of interactive GTTM analyzer and GTTM manual editor.

6. CONCLUSION

We developed a music analyzer called the interactive GTTM analyzer, which derives the grouping structure, metrical structure, time-span tree, and prolongational tree of the GTTM. The analyzer also derives analysis results of chord progression based on the Tonal Pitch Space. The analyzer consists of an automatic GTTM analyzer, called an ATTA, a GTTM manual editor, and a GTTM process editor. By using the process editor, a user can seamlessly change the analysis process of the ATTA and that of the manual editor. The experimental results show that our interactive GTTM analyzer outperformed the GTTM manual editor without an ATTA.

Since the original grouping rules of GTTM are based on monophonic melodies, we have implemented our system faithfully observing the theory. In the future, however, we plan to include harmonic analysis to complement the original theory and to target homophonic music.

7. REFERENCES

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

[2] Hamanaka, M., Hirata, K., and Tojo, S.: "Melody morphing method based on GTTM', *Proceedings of the 2008 International Computer Music conference* (ICMC2008), pp. 155-158, 2008.

[3] Todd, N. "A Model of Expressive Timing in Tonal Music". *Musical Perception*, 3:1, 33-58, 1985.

[4] Widmer, G. "Understanding and Learning Musical Expression", *Proceedings of 1993 International Computer Music Conference* (ICMC1993), pp. 268-275, 1993.

[5] Hirata, K., and Hiraga, R. "Ha-Hi-Hun plays Chopin's Etude", *Working Notes of IJCAI-03 Workshop on Methods for Automatic Music Performance and their Applications in a Public Rendering Contest*, pp. 72-73, 2003.

[6] Hirata, K., and Matsuda, S. "Annotated Music for Retrieval, Reproduction, and Sharing", *Proceedings of 2004 International Computer Music Conference* (ICMC2004), pp. 584-587, 2004.

[7] Hirata, K., and Matsuda, S. "Interactive Music Summarization based on Generative Theory of Tonal Music". *Journal of New Music Research*, 32:2, 165-177, 2003.

[8] Cooper, G., and Meyer, L. B. *The Rhythmic Structure of Music*. The University of Chicago Press, 1960.

[9] Narmour, E. *The Analysis and Cognition of Basic Melodic Structure*. The University of Chicago Press, 1990.

[10] Temperley, D. *The Congnition of Basic Musical Structures*. MIT press, Cambridge, 2001.

[11] Hamanaka, M., Hirata, K., and Tojo, S. "Implementing 'A Generative Theory of Tonal Music'", *Journal of New Music Research*, 35:4, 249-277, 2006.

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

[13] Sakamoto, S., and Tojo, S.: " Harmony Analysis of Music in Tonal Pitch Space", Information Processing Society of Japan SIG Technical Report, Vol. 2009, May 2009 (in Japanese).

[14] PG Music Inc.: Finale, Available online at: http://www.pgmusic.com/finale.htm, 2009.

[15] Recordare, LLC.: MusicXML 2.0 Tutorial, Available online at http://www.recordare.com/xml/musicxml-tutorial.pdf, 2009.

[16] Recordare, LLC.: Dolet 4 for Finale, Available online at http://www.recordare.com/finale/index.html, 2009.

[17] W3C. XML Pointer Language (XPointer). http://www.w3.org/TR/xptr/, 2002.

[18] W3C. XML Linking Language (XLink) Version 1.0. http://www.w3.org/TR/xlink/, 2001.

APPENDIX: PUBLICLY AND DATA FORMAT

We publicize our interactive GTTM analyzer and database of three hundred pairs of scores and correct data at the following URL.

http://music.iit.tsukuba.ac.jp/hamanaka/gttm.htm

We believe that the exhibition of this kind of resource is important for the music information-researching community. The interactive GTTM analyzer is the first application for acquiring time-span trees and

prolongational trees. We hope to benchmark the analyzer to other systems, which will be constructed.

We use the XML as the import and export format since the XML format is extremely qualified to express hierarchical musical structures.

MusicXML

As a primary input format, we chose MusicXML [15] because it provides a common 'interlingua' for music notation, analysis, retrieval, and other applications. For exporting MusicXML from finale we use a plug-in called *Dolet* [16].

GroupingXML

We designed Grouping.XML as an import and export format for hierarchical grouping structures. The GroupingXML has *group*, *note*, and *applied* elements. All *note* elements are inside hierarchical *group* elements. The *applied* elements are located between the end of a group tag and the start of the next group tag, which is where the grouping preference rules (GPRs) are applied. Figure 7a shows a simple example of GroupingXML.

MetricalXML

We designed MetricalXML as an import and export format for metrical structures. MetricalXML has *metric* elements, which require a *dot* attribute, an *at* attribute, and *applied* elements. The *dot* attribute indicates the strength of each beat. The *at* attribute indicates the time from the start of the piece. The *applied* element requires a *level* attribute and a *rule* attribute. In the metrical structure analysis, metrical preference rules (MPRs) are applied to each hierarchy of a dot. The *level* attribute indicates the interval of dots. If there is an onset of a note at the beat position, the *note* element is inserted before the end of the metric element (Figure 7b)

Time-spanXML, ProlongationalXML

The Time-spanXML has *ts*, *primary*, and *secondary* elements. The *ts* element has a *time-span* attribute, a *leftend* attribute, and a *rightend* attribute. Therefore, the *ts* element indicates the length and position of the time-span in a piece of music. In the *ts* element, there is a *head* element, which requires a *note* element indicating the most salient note in the time-span tree. If there is more than one note in the time-span, we can divide the time-span in two parts. One includes the head, and the other 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 7c).

We do not explain ProlongationalXML because its structure is similar to that of the time-span tree.

Tonal Pitch SpaceXML

The Tonal Pitch SpaceXML has *region* elements. Inside the *region* elements there are *chord* elements, and inside the *chord* element there are *note* elements.

Note that *note* elements in GroupingXML, MetricalXML, Time-spanXML, ProlongationalXML, and Tonal Pitch Space-XML are connected to *note* elements in MusicXML using Xpointer [17] and Xlink [18].

Figure 7. GroupingXML, MetricalXML, and Time-spanXML