Proposal for a Performance Data Format

Keiji Hirata[1], **Kenzi Noike**[2] **and Haruhiro Katayose**[2,3]
[1] NTT Communication Science Laboratories
[2] "Intelligent Cooperation and Control," PRESTO/JST
[3] Kwansei Gakuin University
hirata@brl.ntt.co.jp
{noike,katayose}@ksc.kwansei.ac.jp

Abstract

The paper discusses an interchange format for performance data and corpora. We compare the widely accepted XML-based formats for music notation, WEDELMUSIC XML and MusicXML. We propose adopting MusicXML as our interchange format and show an example using it. Last, we suggest a small change to MusicXML for our purposes (MusicXML(4R)) and present a tool for generating MusicXML(4R).

1 Introduction

From our experiences in the Rencon project over the past year, we have come to the conclusion that there is a great need for an interchange format [Selfridge-Field, 1997] for performance data and performance corpora. A performance corpus comprises a score and its real performance data, where real performance data contain deviation information extracted from a human performance. The deviation information means the differences between a neutral (literal, deadpan) performance and an expressive human performance. In a piano performance, the parameters for each note are onset timing, offset timing (or duration), and dynamics (The description is modeled on SMF). The deviation information describes the differences in these parameters.

The reasons an interchange format is needed are concerned with performance-rendering research and how set pieces are presented to contest entrants. We think that performance-rendering research lacks a performance corpus for the development of working systems. While the performance-rendering problem does not always have absolutely correct answers like mathematics, all deviation patterns cannot be accepted. Several methods to synthesize deviation pattens have been proposed. Some try to discover performance-rule sets from a performance corpus, while others employ case-based reasoning using a performance corpus. A performance corpus can also be used for verifying or evaluating performance rendering systems under development. In general, the more performances the corpus contains, the more useful it will be. Moreover, building a large performance corpus requires much effort. Thus, a common interchange format for describing performance data enables many researchers to participate in building the performance corpus and utilize it afterward.

In Rencon, set pieces are limited to Chopin's short piano pieces, which are all already well-known to us. Besides, the titles of the set pieces are presented well in advance of the day of the contest. It turns out that a performance-rendering system can be tuned to specific pieces possibly by introducing ad-hoc rules or manually adjusting the deviations of an individual note. To avoid such artificial tricks, we should introduce a new contest category, in that set pieces are presented to performance rendering systems on the day of a contest so that contest entrants do not know the titles of set pieces beforehand and the total processing time for synthesizing a performance, including both the preprocessing and online phases, is limited. This requires a common interchange format for set pieces that a contest organizer assigns to the systems. The interchange format has to describe both the score and performance data, because some performance rendering systems employ case-based reasoning using performance data or corpus.

The paper summarizes the discussion the Rencon project has had to date regarding an interchange format for performance data and corpora. We propose adopting MusicXML as our interchange format.

This paper is organized as follows. Section 2 explains widely accepted XML-based formats for music notation, WEDELMUSIC XML and MusicXML, and discuss a method using annotation. Section 3 examines ways to describe performance data using MusicXML. Section 4 makes some concluding remarks.

2 XML-based Music Notation Formats

We think that the criteria for a performance data format are simplicity, accuracy, and descriptive power. The format should also be provided with software libraries for programming, tools for editing and long-term organizational support to developers and users. As a result, it is desirable the format is widely available. To meet the need for the libraries, tools, support, and availability, an appropriate format should be selected from existing ones and used with as little modification as possible, hopefully as it is.

Accordingly, we think that XML (Extensible Markup Language) [W3C, 1998] is the most promising format for the time being. XML is a simple, and very flexible text format for exchanging data on the Web and elsewhere. The markup is information inserted into a document used by computers and takes the form of tags inserted into a text to mark its structure. We believe that XML has become so popular mainly because it allows the flexible development of user-defined document types. However, it should also be appreciated that W3C is responsible for XML itself and for the developments of related tools.

The main XML-based music notation formats are WEDELMUSIC XML [Pierfrancesco Bellini and Paolo Nesi, 2001], MusicXML [Michael Good, 2000], MUSITECH [Gieseking and Weyde, 2002], and MusiXML [Castan, 2002]. In terms of describing performance data, we examine WEDELMUSIC XML and MusicXML because they are more developed than others and are coming into wide use.

Both WEDELMUSIC XML and MusicXML are available under a royalty-free license. Commercial and noncommercial organizations and groups provide several dedicated tools and software for them, such as Finale/Sibelius Plug-ins, which enable us to load and store score data in the WEDELMUSIC XML/MusicXML format.

2.1 WEDELMUSIC XML

WEDELMUSIC XML is an XML-compliant format that includes constructs for the description of integrated music objects. WEDEL objects present sections, such as identification, protection, printing, symbolic music (score information), images of music sheets, audio (Wav, MIDI, MP3, etc.), performance, lyrics, and video. Among the sections, the relationship from symbolic music to the audio via performance is important to our purpose, because it can synchronize a note in an audio section with a note of the music score. Hence, the relationship between a note and its deviations can be considered indirect. In WEDELMUSIC XML, each music part of a score is usually described in a separate file. WEDELMUSIC XML does not have a tag for describing pedal position; the positions of the pedals of a piano or a harp are encoded as simple strings (e.g. `HHLH | LHH`).

2.2 MusicXML

MusicXML was designed for a universal translator of common Western musical notation that supports music notation, analysis, information retrieval, and performance applications. The purposes of MusicXML are rather limited than those of WEDELMUSIC XML, and the specifications of MusicXML are more compact. However, a single MusicXML file can convey all information related to a piece and its performance.

Attack and release attributes of the `<note>` tag of MusicXML can be used to describe timing deviations of a note. The following excerpt from MusicXML 0.7 Tutorial [Recordare, 2003] explains this well:

> The duration element should reflect the intended duration, not a longer or shorter duration specific to a certain performance. The note element has *attack* and *release* attributes that suggest ways to alter a note's start and stop times from the nominal duration indicated directly or indirectly by the score.

Attack and release attributes embed timing deviations into the `<note>` tag. One may think that the duration element could be used to describe timing deviations too. However, the duration element is usually used for the consistent mismatch between the performance and notation data across a whole piece, such as the swing feel of equal eighth notes in a jazz tune.

Similarly, for dynamics deviation, dynamics and end-dynamics attributes of the `<note>` tag can be used, where the values given by the attributes correspond to note-on and note-off velocities in SMF, respectively. Hence, in contrast to WEDELMUSIC XML, the relationship between a note and its deviations can be considered direct.

In Figure 1, only attributes related to the note deviations are shown. According to MusicXML 0.7 specifications, the unit of attack and release values is the value of the quarter-note duration divided by the division (This is the same as SMF). In terms of dynamics and end-dynamcs, MusicXML defaults to a MIDI velocity of 90 (roughly a forte), and the dynamics and end-dynamics values are specified as a percentage of the standard MusicXML forte velocity.

```
<note
    dynamics="70"
    end-dynamics="50"
    attack="-20"
    release="+40">
```

Figure 1: Note deviations described by MusicXML

We are, at present, looking favorably at MusicXML, because it can describe deviation data directly within the `<note>` tag and its specifications as a whole are compact yet sufficient for our purpose.

2.3 Annotation Method

Above, we mentioned the intrinsic methods for describing performance data, which are the performance section provided by WEDELMUSIC XML and the `<note>` tag's attributes defined in MusicXML. Now, we can consider another method for describing performance data, in which deviations are described as annotations to the score information in WEDELMUSIC XML or MusicXML.

XPath (XML Path Language) [W3C, 1999] is a language for addressing parts of an XML document. One can access any part of an XML document defined by a given XPath expression. Xpath is also a W3C standard, and a library of standard functions for processing an XPath expression has been published.

Figure 2 outlines the method. For each note, only the deviation data are described in the annotation file. The advantages of this method are: (1) efficiency because a score file is shared, (2) usability of any XML-based music notation formats, and (3) increased flexibility for attaching deviation data. Here, flexibility means that the tempo and division values can be specified separately from a score file itself (described again in Section 3.2), an annotation can be attached to only necessary or desired parts, and annotations to an annotation are also allowed.

On the other hand, the disadvantages are the needs for an overhead to maintain the coherence of XPath references among separate files spread over the WWW, and vulnerability to modifications to score and annotation files because

```
<sound tempo="120"/>
<divisions>192</divisions>
<deviations>
    <deviation
        noteref="an XPath expression to
                 a corresponding note
                 in an XML file"
        dynamics="70"
        end-dynamics="50"
        attack="-20"
        release="+40">
    <deviation
        noteref=".. to another note .."
        ...>
    ...
</deviations>
```

Figure 2: Note deviations described by annotation

these files are not immutable in general. We will be able to cope with the first problem by constructing the support software/tools. In terms of the second, by using MD5 checksum, for example, one can check if score and annotation files have been modified unexpectedly. However, there are no standard techniques to recover lost original files.

3 Performance Description by MusicXML

Since our purpose at present is to distribute performance data to contest entrants, we want to adopt a method that uses the attributes related to the note deviations of MusicXML (Section 2.2).

3.1 Suggested Change to MusicXML

We show a simple example in Figure 3, where (a) depicts a sample melody of three notes "G E E" and the graph in (b) depicts the deviations of the three notes. In the graph, the onset and offset timings on the x-axis and dynamics on the y-axis. Below the graph, the corresponding MusicXML description (partwise) is shown. For example, the third note (half note of pitch E4) has the dynamics, attack, and release values of 112, +30, and -46, respectively.

We have already made a suggested change to MusicXML. As mentioned in Section 2.2, the dynamics and end-dynamics values are specified as a percentage of 90 (the default MIDI velocity of MusicXML). However, here, the velocity values of SMF themselves are assigned to the dynamics and end-dynamics values for compliance with SMF. If we had followed the MusicXML specification, we would have had to write `dynamics="124.444"`, since 112/90 = 1.2444· · ·.

In terms of attack and release values, since the division value is 240 and tempo is 120, 30 means one-eighth of a quarter note, i.e., 32nd note and 62.5 `ms`. Similarly, 46 means 95.83 `ms`.

3.2 Matching Tool

The Rencon project now plans to provide a matching tool that takes as input score data in MusicXML and the corresponding performance data in SMF and generates a MusicXML file containing the deviation information (We call this MusicXML *MusicXML(4R)*, where 4R stands for "for Rencon") (Figure 4). To generate MusicXML(4R), for each note in an

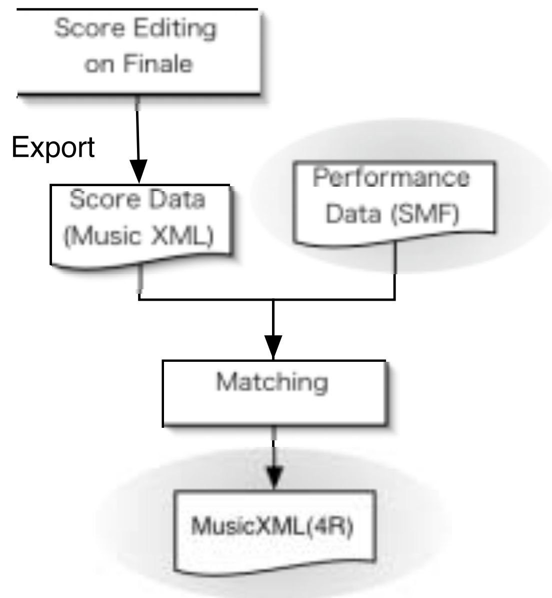


Figure 4: Flow Diagram of Generating MusicXML(4R)

input MusicXML file, the tool first looks for a corresponding note in SMF. Since this matching method preserves the order of all notes in MusicXML, if there are more than one performance for a piece (a MusicXML file and several SMF files), the order of all notes occurring in MusicXML(4R) is always identical.

Here, we mention a small inconvenience of the Finale Plug-in for MusicXML. When one stores a score file in MusicXML using Finale, the Finale Plug-in for MusicXML generates the division value after it takes into account all note durations in a part. For example, suppose there are eighth notes and quarter notes in a part. Then, the division value is set to 2, because if the half-duration of a quarter note is regarded as a unit, the durations of all notes within the part can be described. That is, the greatest common divisor (GCD) of 1/2 and 1 is 1/2, although this calculation does not follow the correct definition of GCD. For another example, suppose there are eighth notes and triplets of quarter notes in a part. Then, the division value is set to 6, because GCD of 1/2 and 1/3 is 1/6. Therefore, we cannot use the division value generated by the Finale Plug-in for MusicXML as a unit for the values of the attack and release attributes.

Therefore, our matching tool first copies the division value from SMF into that of MusicXML(4R) and converts the duration value of every note in accordance with the unit of the SMF's division.

In terms of the unit of the division value in MusicXML(4R), since we are thinking of handling audio signal as performance data as well in the future, a microsecond may be convenient as the unit.

4 Concluding Remarks

We have considered a few possibilities for a performance data format based on XML, proposed the adoption of MusicXML and suggested a small change to it (MusicXML(4R)). We recognize that, for dissemination of MusicXML(4R), it is quite important to develop and support substantial tools and to construct a performance corpus of high quality. Moreover, we would like to deploy an annotation method in future. We hope that this discussion serves as a starting point for developing a useful performance corpus not only for performance rendering but also the AI and music research community.

Acknowledgment: This material is partly supported by "Intelligent Cooperation and Control," PRESTO/JST. We would like to thank Ms. Mitsuyo Hashida for her suggestive, analytic comments on the MusicXML specification.

References

[Castan, 2002] Gerd Castan. MusiXML - XML music notation file format. http://www.music-notation.info/en/musixml/MusiXML.html, 2002.

[Gieseking and Weyde, 2002] Martin Gieseking and Tillman Weyde. Concepts of the MUSITECH Infrastructure for Internet-Based Interactive Musical Applications. In *Proceedings of Second Int'l Conf. on Web Delivering of Music*, 2002.

[Michael Good, 2000] Michael Good. Representing Music Using XML. In *Proceedings of ISMIR 2000*, 2000.

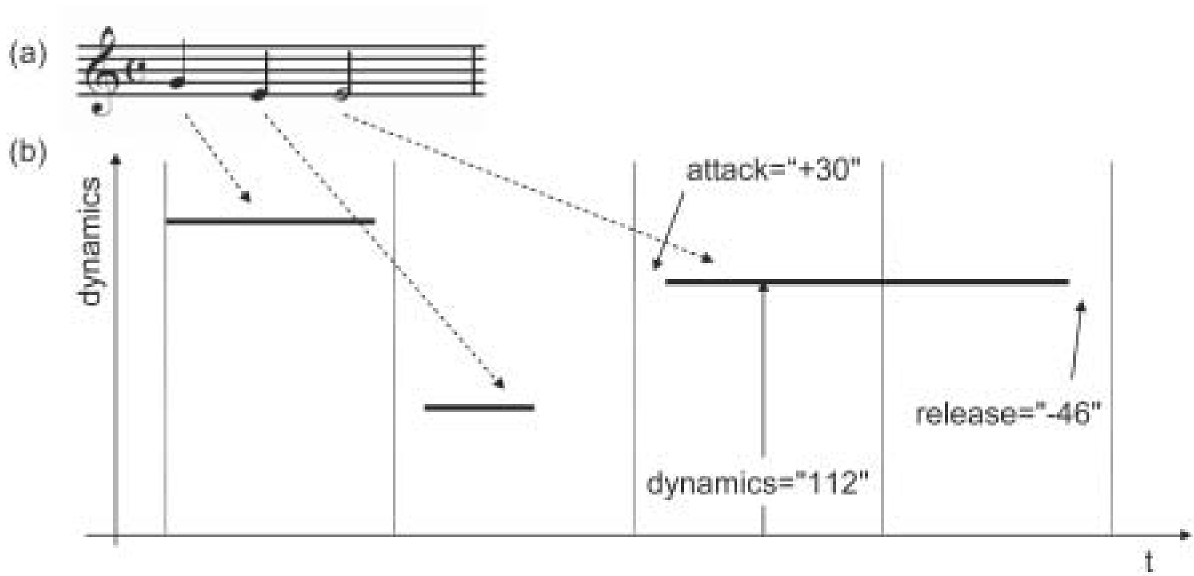
[Pierfrancesco Bellini and Paolo Nesi, 2001] Pierfrancesco Bellini and Paolo Nesi. WEDELMUSIC Format: an XML Music Notation Format for Emerging Applications. In *Proceedings of First Int'l Conf. on Web Delivering of Music*, 2001.

[Recordare, 2003] Recordare. MusicXML 0.7 Tutorial. http://www.recordare.com/xml/musicxml-tutorial.pdf, 2003.

[Selfridge-Field, 1997] Eleanor Selfridge-Field. *Beyond MIDI*. The MIT Press, 1997.

[W3C, 1998] W3C. Extensible Markup Language (XML). http://www.w3.org/XML/, 1998.

[W3C, 1999] W3C. XML Path Language (XPath). http://www.w3.org/TR/xpath, 1999.




```xml
<part id="P1">
  <measure number="1">
    <attributes>
      <divisions>240</divisions>
      <key>
        <fifths>0</fifths>
        <mode>major</mode>
      </key>
      <time symbol="common">
        <beats>4</beats>
        <beat-type>4</beat-type>
      </time>
      <clef>
        <sign>G</sign>
        <line>2</line>
      </clef>
    </attributes>
    <sound tempo="120"/>
    <note dynamics="123" attack="0" release="-12">
      <pitch>
        <step>G</step>
        <octave>4</octave>
      </pitch>
      <duration>240</duration>
      <voice>1</voice>
      <type>quarter</type>
      <stem>up</stem>
    </note>
    <note dynamics="52" attack="+25" release="-85">
      <pitch>
        <step>E</step>
        <octave>4</octave>
      </pitch>
      <duration>240</duration>
      <voice>1</voice>
      <type>quarter</type>
      <stem>up</stem>
    </note>
    <note dynamics="112" attack="+30" release="-46">
      <pitch>
        <step>E</step>
        <octave>4</octave>
      </pitch>
      <duration>480</duration>
      <voice>1</voice>
      <type>half</type>
      <stem>up</stem>
    </note>
  </measure>
</part>
```

Figure 3: Deviations of a simple melody “G E E”