USING XML TECHNOLOGIES IN MUSIC

Gilbert Kuipers
Department of Science
Valley City State University
Valley City, ND 58072
Phone: 701-845-7324
Gilbert.Kuipers@.vcsu.edu

Michael Good
Recordare LLC
P.O. Box 3459
Los Altos, California 94024
Phone: 650-965-7267
good@recordare.com

Abstract

For a variety of reasons Extensible Markup Language (XML) is rapidly emerging as a universal data standard. It is not surprising that there have been many attempts to use XML (and the earlier non-XML SGML) technologies for multimedia and for music [1]. Unfortunately there are many complex problems in the integration of music, music notation, and multimedia. Anyone can create a new XML schema or DTD. The difficulties are limiting the problem, developing a practical solution platform, implementing this solution in various applications, and having it selected by users who are often not very computer literate. If we limit the problem to an interchange format for notation, analysis, retrieval, and performance applications, MusicXML is currently the only XML language which has emerged as a practical format for use by real musicians [2].

This paper will begin with a brief overview of various data formats used for common Western music and the need for the new music interchange format. The design and use of MusicXML will be presented. Examples will show the creation of MusicXML files by entering the data directly or by scanning printed paper scores, along with the conversion of data between formats. Since XML data is tagged text data, MusicXML data can easily be analyzed, e.g. to determine the percentage of Bach's pieces which are quarter notes or the number of C major chords in a symphony. MusicXML is suitable for data archiving. Project Gutenberg, the Internet's oldest producer of free electronic books, has approved MusicXML as the only non-proprietary text-based format approved for music submissions. MusicXML attempts to provide an interchange language that is well designed from musical, human, and computer perspectives. It does not solve all problems, but augments other specialized proprietary music formats available today. It should also be possible to integrate MusicXML with other XML technologies, such as SVG, SMIL, and someone's future development of a non-Western XML music DTD or schema.

Introduction

Common Western music notation is a symbolic method of representing music for performers and listeners. It is the most well known form for musical notation. Your sheet music will almost certainly be in this form if you play a band instrument, play the piano in church, or sing in the chorus. This notation can be encoded in a graphical image or in the Portable Document Format (PDF), but these contain no musical semantics. Various companies like Sunhawk, Finale, Noteheads, and Sibelius all have different proprietary music formats for their sheet music. These generally need a free viewer to display, play and print the files. Some of these companies have music notation software which will allow one to modify the files. But converting data from one program to another is difficult. There is very little additional value to purchasing this music in this format as opposed to printed music on paper.

Many computer users today are aware of MP3, Quicktime, and other music similar formats. Although these are excellent for playing recorded music, they contain no information about the notes played.

MIDI (Musical Instrument Digital Interface format) was developed in the early 1980's as a music interchange format between electronic musical instruments and various accessories of various vendors. MIDI has been incredibly successful. It is truly a universal standard found today in almost every electronic music instrument. However MIDI suffers from the limitations of computers of the late 1970's. MIDI doesn't know the difference between a C-sharp and a D-flat. It doesn't represent various details found on printed scores, such as stem direction, repeats, measures, and slurs.

Various music interchange formats have been developed in the past couple of decades, but none have been as successful as MIDI. Notation Interchange File Format (NIFF) was developed to interchange music between scanning and notation applications. NIFF is better than MIDI as a graphical representation, but inferior for performance and analysis. Standard Music Description Language (SMDL) was designed to be a general purpose format based on Standard Generalized Markup Language (SGML). Unfortunately both SMDL and SGML are complex and difficult to understand. A subset of SGML, called XML or Extensible Markup Language, keeps most of the useful parts of SGML and discards much of its complexity.

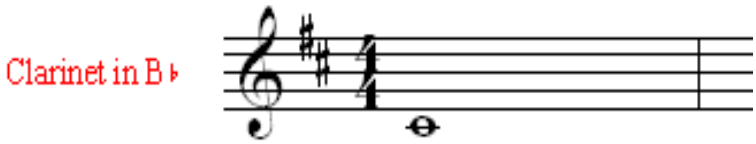
MusicXML

MusicXML is an attempt to use modern technology to do for online sheet music and music software what MIDI did for electronic musical instruments. If we limit the problem to an interchange format for notation, analysis, retrieval, and performance applications, MusicXML is currently the only XML language which has emerged as a practical format for use by real musicians. It is a universal translator for common Western music notation from the 17th century onwards. It augments, but does not replace, existing specialized formats of individual specialized applications.

MusicXML is available under a royalty-free license from Recordare, which is similar to a license from the World Wide Web Consortium (W3C). One advantage of XML is that the files can be edited in any standard word processor. There is also a standard process of converting from one application to another, i.e. it should be easy to convert music data files from MusicXML to some other XML music format. Much more background information about Music and XML can be found at the Cover Pages Web site [1] and the Recordare Web site [2].

Let's look at a trivial one-note song in 4/4 time as shown in Figure 1. This was created using Finale 2002 notation software. Although I called it Middle C, it really represents Middle C#. At this point the song could be printed as normal sheet music. It could be saved for the Web, but would only be viewable with the SmartMusic Viewer. It could be saved in a limited number of proprietary formats or as a MIDI file.

Figure 1: Middle C



The MusicXML version of that is listed below in Figure 2. This was created by exporting the file using the Dolet plug-in. Persons not familiar with XML will quickly notice the large file size typical of XML files. This is not a big of problem today as it would have been several years ago. If file size is a problem for sending a file over the internet, then compression methods can be used. Compressed MusicXML files are roughly the same size as Standard MIDI files for the same music, and contain much more notation information.

Figure 2: Middle C Listing

```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE score-partwise PUBLIC "-//Recordare//DTD MusicXML 1.0
Partwise//EN" "http://www.musicxml.org/dtds/partwise.dtd">
<score-partwise>
      <movement-title>Middle C</movement-title>
      <identification>
            <creator type="composer">Gilbert</creator>
            <rights>none</rights>
            <encoding>
                  <software>Finale 2002 for Windows</software>
```

```
                  <software>Dolet for Finale 1.3.1</software>
                  <encoding-date>2004-03-12</encoding-date>
            </encoding>
      </identification>
      <part-list>
            <score-part id="P1">
                  <part-name>Clarinet in Bb</part-name>
                  <score-instrument id="P1-I1">
                        <instrument-name>Clarinet</instrument-name>
                  </score-instrument>
                  <midi-instrument id="P1-I1">
                        <midi-channel>1</midi-channel>
                        <midi-program>72</midi-program>
                  </midi-instrument>
            </score-part>
      </part-list>
      <part id="P1">
            <measure number="1">
                  <attributes>
                        <divisions>1</divisions>
                        <key>
                              <fifths>2</fifths>
                              <mode>major</mode>
                        </key>
                        <time>
                              <beats>4</beats>
                              <beat-type>4</beat-type>
                        </time>
                        <clef>
                              <sign>G</sign>
                              <line>2</line>
                        </clef>
                        <transpose>
                              <diatonic>-1</diatonic>
                              <chromatic>-2</chromatic>
                        </transpose>
                  </attributes>
                  <sound tempo="120"/>
                  <note>
                        <pitch>
                              <step>C</step>
                              <alter>1</alter>
                              <octave>4</octave>
                        </pitch>
                        <duration>4</duration>
                        <voice>1</voice>
                        <type>whole</type>
                  </note>
            </measure>
      </part>
</score-partwise>
```

Notice the rich amount of information provided in the file. Everything needed is provided for printing the score. The file is a text file, one that can be pasted into Microsoft Word or any other text editor.

MusicXML can represent scores either partwise (measures within parts) or timewise (parts within measures), with XSLT stylesheets to go between the two. It is represented here as partwise. This example uses a partwise score for the Bb Clarinet. The midi instrument id is P1-I1.

The attributes element contains musical attributes of a score. Although the key of C has no sharps or flats, I accidentally included two sharps, F and C, which is indicated by the circle of fifths. There are 4 beats in a measure and a quarter-note (beat-type) gets one beat. Treble clef or G clef is positioned on the second line. The Bb clarinet is a transposing instrument. C# as written is really B natural, which is 2 steps down on the chromatic scale and one step down on a diatonic scale.

MusicXML files can also be created from printed music scores. Figure 3 below is the beginning of the Mason's Apron as Scanned on a flatbed scanner. Figure 4 below is the same section as converted to a music score using the SharpEye 2 program. This is essentially an OCR program. All OCR programs make mistakes. The program did not notice that the first three notes form a triplet. But this is faster than entering the data by hand as in figure 1. The mistakes can be edited in a program like Finale. Figure 5 contains the first two measures as converted to MusicXML.

Figure 3: Mason's Apron as Scanned

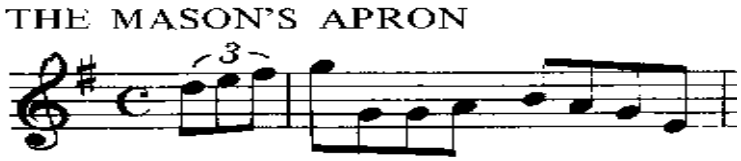


Figure 4: Mason's Apron as Converted to Music

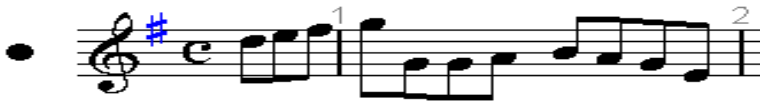


Figure 5: Mason's Apron as MusicXML

```
<part id="P1">
            <measure number="1">
                  <attributes>
                        <divisions>6</divisions>
```

```
                        <key>
                              <fifths>1</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>
                        <pitch>
                              <step>D</step>
                              <octave>5</octave>
                        </pitch>
                        <duration>3</duration>
                        <voice>1</voice>
                        <type>eighth</type>
                        <stem>down</stem>
                        <beam number="1">begin</beam>
                  </note>
                  <note>
                        <pitch>
                              <step>E</step>
                              <octave>5</octave>
                        </pitch>
                        <duration>3</duration>
                        <voice>1</voice>
                        <type>eighth</type>
                        <stem>down</stem>
                        <beam number="1">continue</beam>
                  </note>
                  <note>
                        <pitch>
                              <step>F</step>
                              <alter>1</alter>
                              <octave>5</octave>
                        </pitch>
                        <duration>3</duration>
                        <voice>1</voice>
                        <type>eighth</type>
                        <stem>down</stem>
                        <beam number="1">end</beam>
                  </note>
            </measure>
            <measure number="2">
                  <note>
                        <pitch>
                              <step>G</step>
                              <octave>5</octave>
                        </pitch>
                        <duration>3</duration>
                        <voice>1</voice>
```

```
                        <type>eighth</type>
                        <stem>down</stem>
                        <beam number="1">begin</beam>
                  </note>
                  <note>
                        <pitch>
                              <step>G</step>
                              <octave>4</octave>
                        </pitch>
                        <duration>3</duration>
                        <voice>1</voice>
                        <type>eighth</type>
                        <stem>down</stem>
                        <beam number="1">continue</beam>
                  </note>
                  <note>
                        <pitch>
                              <step>G</step>
                              <octave>4</octave>
                        </pitch>
                        <duration>3</duration>
                        <voice>1</voice>
                        <type>eighth</type>
                        <stem>down</stem>
                        <beam number="1">continue</beam>
                  </note>
                  <note>
                        <pitch>
                              <step>A</step>
                              <octave>4</octave>
                        </pitch>
                        <duration>3</duration>
                        <voice>1</voice>
                        <type>eighth</type>
                        <stem>down</stem>
                        <beam number="1">end</beam>
                  </note>
                  <note>
                        <pitch>
                              <step>B</step>
                              <octave>4</octave>
                        </pitch>
                        <duration>3</duration>
                        <voice>1</voice>
                        <type>eighth</type>
                        <stem>up</stem>
                        <beam number="1">begin</beam>
                  </note>
                  <note>
                        <pitch>
                              <step>A</step>
                              <octave>4</octave>
                        </pitch>
                        <duration>3</duration>
                        <voice>1</voice>
                        <type>eighth</type>
                        <stem>up</stem>
```

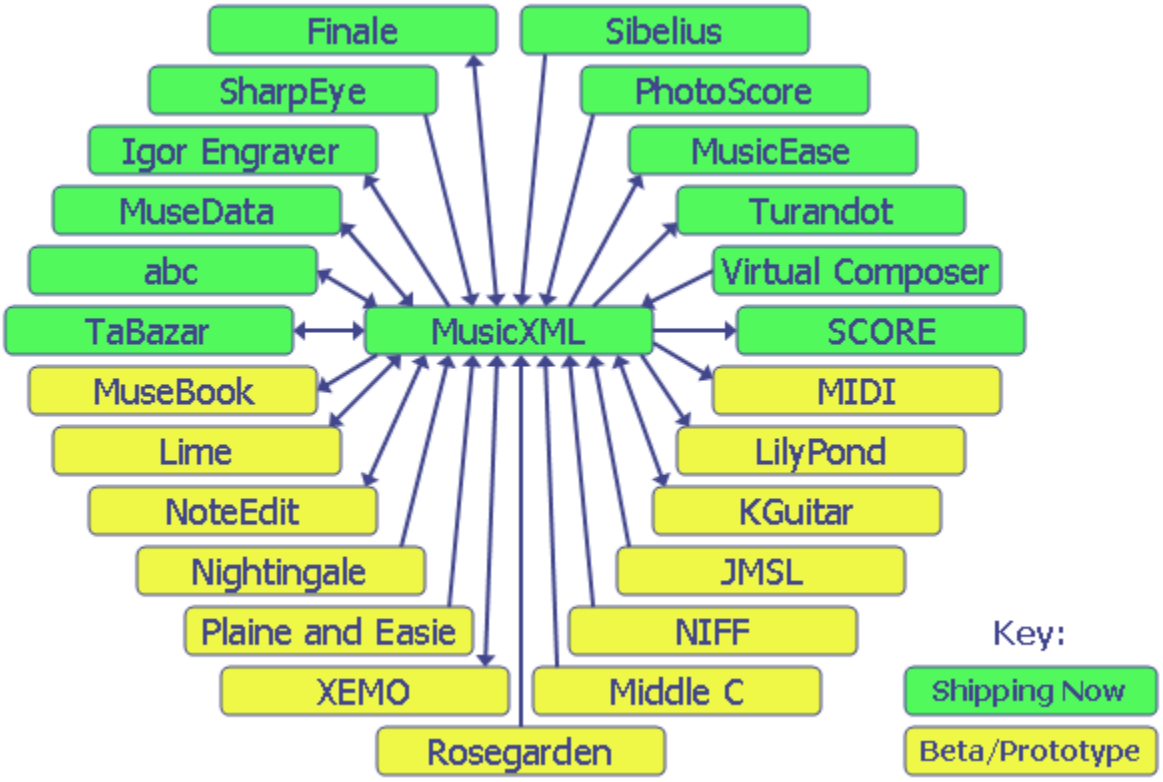
```
                        <beam number="1">continue</beam>
                  </note>
                  <note>
                        <pitch>
                              <step>G</step>
                              <octave>4</octave>
                        </pitch>
                        <duration>3</duration>
                        <voice>1</voice>
                        <type>eighth</type>
                        <stem>up</stem>
                        <beam number="1">continue</beam>
                  </note>
                  <note>
                        <pitch>
                              <step>E</step>
                              <octave>4</octave>
                        </pitch>
                        <duration>3</duration>
                        <voice>1</voice>
                        <type>eighth</type>
                        <stem>up</stem>
                        <beam number="1">end</beam>
                  </note>
```

MusicXML can also be created by playing music on a keyboard with a MIDI output. But MIDI has no discrete note element; rather, notes are bounded by NoteOn and NoteOff elements. MIDI doesn't distinguish between D-sharp and E-flat since both are the same note on the keyboard. MIDI has no representation for rests. They are understood by the absence of notes. Both Sibelius and Finale can import MIDI files and convert them into sheet music. Unfortunately these programs have to guess and often guess differently.

Present Status

Figure 6 shows the present status of software programs which will import and/or export MusicXML. For example, if you are using Finale and a friend is using Sibelius, then you can exchange MusicXML files. Although there are several other efforts in using XML to encode music [1], MusicXML is undoubtedly the most mature and the most implemented.

Figure 6: Present Status of Software Programs [2]



Future Directions

Since XML data is tagged text data, MusicXML data can easily be analyzed, e.g. to determine the percentage of Bach's pieces which are quarter notes or the number of C major chords in a symphony. MusicXML is suitable for data archiving. Project Gutenberg, the Internet's oldest producer of free electronic books, has approved MusicXML as the only non-proprietary text-based format approved for music submissions. MusicXML attempts to provide an interchange language that is well designed from musical, human, and computer perspectives. It does not solve all problems, but augments other specialized proprietary music formats available today. It should also be possible to integrate MusicXML with other XML technologies, such as SVG, SMIL, and someone's future development of a non-Western XML music DTD or schema.

References

1. There are now numerous references on XML. The Cover Pages web site at **http://xml.coverpages.org/** is one of the most complete sources of information about XML. The section at **http://xml.coverpages.org/xmlMusic.html** describes various initiatives involving music.

2. The Recordare web site at **http://www.musicxml.org/xml.html** is the most complete source of information about MusicXML.