Chapter 9
MusicXML: The First Decade

Michael D. Good
MakeMusic, Inc., USA

ABSTRACT

MusicXML is a universal interchange and distribution format for common Western music notation. MusicXML’s design and development began in 2000, with the purpose to be the MP3 equivalent for digital sheet music. MusicXML was developed by Recordare and can represent music from the 17th century onwards, including guitar tablature and other music notations used to notate or transcribe contemporary popular music. MusicXML is supported by over 160 applications. The development and history of MusicXML is described in this chapter.

PURPOSE

MusicXML is a universal interchange and distribution format for common Western music notation. It is designed so that a single XML file can be used by a wide range of music software applications. These applications include the display and editing of music notation, the playback and editing of a performance of a musical score, musical analysis, and music retrieval.

MusicXML’s design and development began in 2000, when Internet audio applications were coming into widespread use. Digital sheet music retail sites like Sunhawk were also starting to appear. However, there was no standard format for music notation that compared to the MP3 format for audio. Instead, each application had its own proprietary format. The only way to share sheet music files between applications was with Standard MIDI Files. But MIDI loses a great deal of information when representing music notation (Selfridge-Field, 1997). MusicXML was designed to be the MP3 equivalent for digital sheet music.

MusicXML represents music from the 17th century onwards, including guitar tablature and other music notations used to notate or transcribe contemporary popular music. Since MusicXML was being developed by Recordare, a commercial company, the focus was relentlessly practical. The format needed to support a wide range of Western music of commercial and artistic interest, in a way that was practical for contemporary music notation applications to use.

DOI: 10.4018/978-1-4666-2497-9.ch009
MusicXML does not represent earlier Western music in its original notation, nor does it represent non-Western music. Most performers who read Western sheet music cannot read early or non-Western music. MusicXML reflects these commonalities and boundaries of musician experience. This makes the markup much more musician-friendly than is possible in more abstract, general-purpose languages. This is important for getting a format widely implemented, as most music notation software developers are musicians themselves (Good, 2006a). Over-abstraction was a fundamental reason why the earlier Standard Music Description Language (SMDL) format (Sloan, 1997) was never adopted.

**DESIGN APPROACH**

MusicXML models common Western music notation using concepts and vocabulary that are familiar to Western musicians. The basic organization of musical structure comes from the MuseData format (Hewlett, 1997), with additional ideas from the Humdrum format (Huron, 1997). MusicXML extended these designs to support popular as well as classical music, including tablature, guitar chord diagrams, and percussion notation.

Our initial implementations confirmed that MuseData was indeed a good starting point for MusicXML’s design. The basic musical organization mapped smoothly to the music representation of popular music notation editors such as Finale and Sibelius. MusicXML’s element and attribute names are based on the English-language musical terms used in the USA. This makes the format easier to read and understand than those that use either non-musical names or cryptic abbreviations, making the format better suited for archival use.

MusicXML primarily models a document—a musical score—rather than an abstraction of a document. Different aspects of music—the musical structure, the appearance of a particular score engraving, and the interpretive details of a particular musical performance—are all included in a single unified document, mirroring how contemporary music software works. The integration of appearance is especially important, since appearance conveys semantic meaning in musical scores. MusicXML places the musical data in XML elements, and places the visual and performance information (based on MuseData’s print and sound suggestions) in XML attributes or special-purpose XML elements. This element/attribute distinction follows commonly accepted best practices for XML language design (Harold, 2004).

To illustrate, Box 1 shows how MusicXML can represent a staccato middle C. The formatting data assumes standard positioning in the treble clef, and the duration value assumes a definition of 4 divisions per quarter note.

Note how basic musical concepts are represented as elements, with print and sound suggestions represented as attributes. Since what is notated does not always match what is performed, MusicXML has separate elements to capture this duality. For instance, the <type> element represents what is notated, while the <duration> element represents what is performed. The default-\(y\) attribute for the stem element indicates that the upstem should end one and a half spaces below the top line of the staff. The release attribute for the note element indicates that the played duration should be one division shorter than indicated by the value in the duration element—equivalent to shortening by a sixteenth note in this example.

MusicXML’s approach based on musical semantics differs dramatically from the approach of the earlier Notation Interchange File Format (NIFF—Grande, 1997). NIFF’s graphical format primarily defines notes by their placement on the staff rather than by their musical pitch. This makes NIFF difficult to match to the data structures of contemporary notation programs, or any programs that need to play music as well as display it (Good, 2006b). NIFF’s adoption was thus largely limited.