DAISY Music Braille Project:

Suggestions for MusicXML 3.2 and MNX Implementation

Haipeng Hu, 2 February 2019

This document provides general suggestions on the implementation of MusicXML 3.2 and MNX formats, hoping they can be more braille-transcription-friendly. But it doesn't limit to the requirement of braille music transcription. Some of the points may also benefit the common transformation of musical data to be more accurately and clearly.

# I. MusicXML 3.2 Suggestions

1. If possible, implement a flow to include all system-wide or page-wide texts, lines and symbols, like the "global" section of MNX, so that all things in it can affect the whole score instead of the topmost staff or (in orchestra) piccolo and first violin staves. Their positions can be placed using the forward command.

2. Create a category attribute for words and lines, to include both default and user-defined categories of texts and lines, so that the braille transcription software can determine how to convert them in braille (correct equivalents and positions according to braille music convention).

3. Create non-existing text categories like string numbers, positions (these two are exclusively for bowed string instruments, not tablature scores), fingering (not the standard fingering element, for sometimes the engraver may enter finger numbers as texts in various reasons), PIMA etc. These are intended for braille transcription use, for these Roman or Arabic numbers have great ambiguity, and can only be put as words in notation softwares. When we apply correct categories to them, they can then be transcribed as correct braille equivalents.

4. Besides common time signatures including numerator(s) and denominator(s), also support special time signatures represented by note(s) such as a single dotted quarter, or like O Fortuna of Carmina Burana, 3 whole notes.

5. Create more complicated ornaments elements met in Bach's scores, like various extended mordents, trills and turns, schleifer line, tremblement, bebung etc. many of them can be found in the symbols gallary in Sibelius.

6. If an ornament contains accidentals, they should be put together with the ornament. The accidentals should have position indications to show whether they are above or below the sign. This is extremely important for all turns. Although we can show their positions visually using axis values like in MusicXML, these written-out attributes will be much more clear, especially for braille transcription.

7. For arpeggiated lines, include an attribute to indicate whether it's for two staves, thus a connect-staff arpeggio. A length value doesn't make any sense for braille transcription. Although the software may not export this attribute automatically (if the developers don't change the behavoir of the software), we can still manually markup it with the attribute to get correct braille signs.

8. Add way to express multiple staccato dots above a tremolo. These can be done using symbols in Sibelius, but there are no MusicXML equivalents for them.

9. Support laissez-vibrer, to avoid using tie without target note. This also makes braille conversion easier and clearer.

10. For lines, simple "bracket" or "dashes" is not enough. it's better to define an exclusive line element for staff and system lines, with dashed or solid as attributes. If a user-drawn line is not a straight line, more intermediate points should be inserted into the sequence to describe the explicit curve points. If possible, marking the points with optional staff position (approximate pitch location) may be useful for braille transcription, since braille is one-dimension only, and we need a pitch position to know where the line goes to or passes. This is useful for representing curve lines and glissando lines without destination note but just lines pointing to a place on the staff (this may need corresponding algorithm on the notation software side, but we should reserve this feature for possible use).

# II. MNX Suggestions

The following suggestions to MNX will only focus on CWMNX, since GMNX scores are graphical, and are impossible to be converted into braille. Sometimes there are questions left for the developers to think of a better solution.

1. Generally, include all symbols, articulations, ornaments, chords, diagram expressions, pictograms and other existing features from MusicXML 3.1.

2. Have a good strategy to include both default and user-defined categories of texts and lines, so that the braille transcription software can determine how to convert them in braille (correct equivalents and positions according to braille music convention). If a line contains text or symbol, it's better to include the text or symbol as attributes directly in the line element itself, rather than creating elements apart from the line.

3. Create non-existing text categories like string numbers, positions (these two are exclusively for bowed string instruments, not tablature scores), fingering (not the standard fingering element, for sometimes the engraver may enter finger numbers as texts in various reasons), PIMA etc. These are intended for braille transcription use, for these Roman or Arabic numbers have great ambiguity, and can only be put as words in notation softwares. When we apply correct categories to them, they can then be transcribed as correct braille equivalents.

4. Support more widely defined dynamic marks such as sfff, not limited by SMUFL-defined ones. These characters are usually entered as dynamic texts, so simply include them as dynamic marks other than using the unrecognized tags like other-dynamics in MusicXML.

5. Since MNX has no division values like MusicXML, when different time signatures are applied to different parts and they share the same bar length with different scaled durations, every individual <global> element should specify appropriate duration scaling to make the affected parts take the correct bar length. This can avoid using large number of time-modifications or hidden tuplets throughout the music, and makes the output much cleaner.

6. Besides common time signatures including numerator(s) and denominator(s), also support special time signatures represented by note(s) such as a single dotted quarter, or like O Fortuna of Carmina Burana, 3 whole notes.

7. Besides common metronome marks of tempo bpm, also support unusual metronome marks. Inherit the already-complete specification from MusicXML.

8. Support most of the complicated ornaments met in Bach's scores, like various extended mordents, trills and turns, schleifer line, tremblement, bebung etc. many of them can be found in the symbols gallery in Sibelius.

9. If an ornament contains accidentals, they should be put together with the ornament. The accidentals should have position indications to show whether they are above or below the sign. This is extremely important for all turns. Although we can show their positions visually using axis values like in MusicXML, these written-out attributes will be much more clear, especially for braille transcription.

10. For arpeggiated lines, include an attribute to indicate whether it's for two staves, thus a connect-staff arpeggio. A length value doesn't make any sense for braille transcription. Although the software may not export this attribute automatically (if the developers don't change the behavoir of the software), we can still manually markup it with the attribute to get correct braille signs.

11. Add way to express multiple staccato dots above a tremolo. These can be done using symbols in Sibelius, but there are no MusicXML equivalents for them.

12. Support laissez-vibrer, to avoid using tie without target note. This also makes braille conversion easier and clearer.

13. For symbol element, it's better not to limit the symbol values to SMUFL glyphs. Some user-defined symbols and symbols from Sibelius will have their own names. Should we create another kind of symbol element to include these?

14. For lines, simple "bracket" or "dashes" is not enough. it's better to define an exclusive line element for staff and system lines, with dashed or solid as attributes. If a user-drawn line is not a straight line, more intermediate points should be inserted into the sequence to describe the explicit curve points. If possible, marking the points with optional staff position (approximate pitch location) may be useful for braille transcription, since braille is one-dimension only, and we need a pitch position to know where the line goes to or passes. This is useful for representing curve lines and glissando lines without destination note but just lines pointing to a place on the staff (this may need corresponding algorithm on the notation software side, but we should reserve this feature for possible use).