MusicXML

Amy Ruskin
MUMT 621

What is MusicXML?

- Invented by Michael Good of Recordare LLC
- Introduced at ISMIR 2000
- XML-based format for (Western) musical notation
- De facto interchange standard
- Used by music **publishers**, **composers**, and **researchers** in software applications for musical **notation**, **performance**, **analysis**, and **retrieval**

(Good 2001a; British Library 2018)

Background

- No standard format for notation in early digital sheet music marketplace (Good 2001b; Good 2013)
  - PDF most common, but no musical semantics embedded in file
  - Formats from Sunhawk, MusicNotes, Sibelius, Noteheads, etc. required proprietary software to play, view, and print files
- Importance of interchange between formats
  - Different software used in different steps of composition and distribution processes—e.g. composer using Sibelius, arranger using Finale (Good 2006b)
- Previous attempts at developing music interchange formats largely unsuccessful (Good 2001b)

Can’t they just use MIDI?

- As of 2001, MIDI was the only widely used symbolic music interchange format (Good 2001b)
- MIDI’s intended use was communication between electronic instruments → shortcomings when used for music notation (Good 2001b; Good 2013)
  - No discrete note elements
  - No representation of rests
  - No way to distinguish between enharmonics
  - No explicit representation of beams, stem direction, slurs, clefs, and more
- Information lost when transferring files using MIDI as interchange format (Good 2006b)

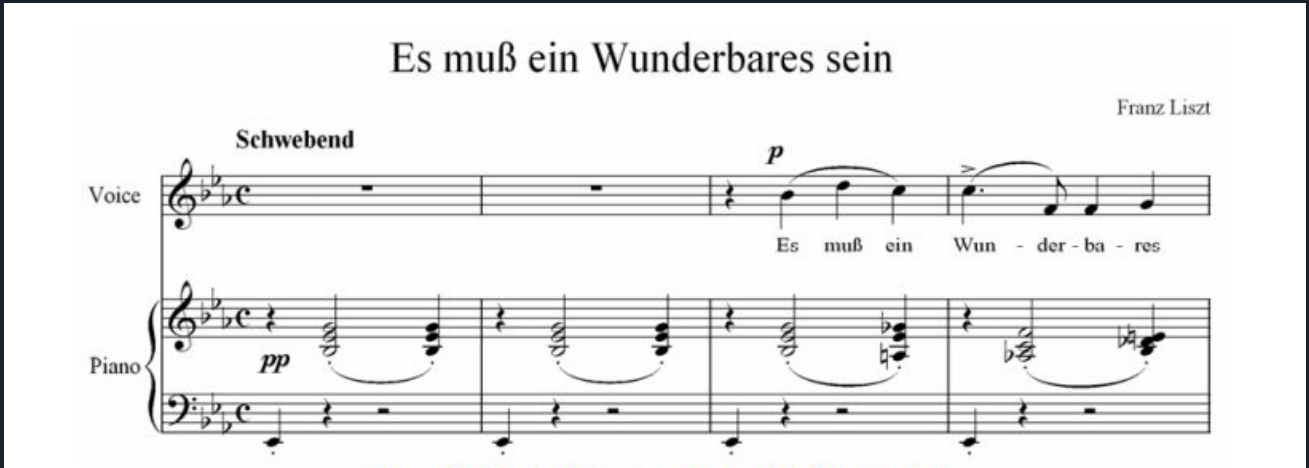


Figure 1. Original music as entered in Sibelius 4.1 (Good 2006b)

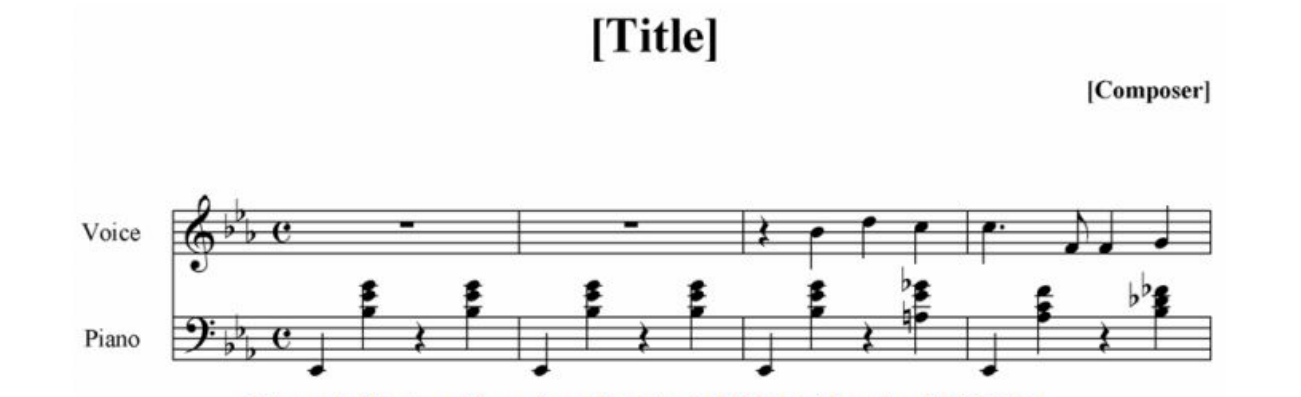


Figure 2. Music as transferred to Finale 2007 via Standard MIDI file (Good 2006b)

Limitations of Other Predecessors

- Standard Music Description Language (SMDL)
  - Scope too broad—tried to represent all music from all time periods
  - Esoteric terminology—e.g. *cantus*, *gamut*
  - Adopted as ISO standard, but not heavily implemented (Good 2006b)
- Notation Interchange File Format (NIFF)
  - Graphical format—notes defined by placement on staff rather than musical pitch
  - Useful as export format from music scanning programs, but difficult to use in notation or playback software (Good 2006b)
  - Bug in Sibelius's NIFF importer: not checking for accidentals in tied notes (Good 2006a)

MusicXML Design Principles

- Based on MuseData and Humdrum (academic music formats)
  - Initial version mostly adapted MuseData format into XML
- **Limited scope:** common Western musical notation from 17th century onwards (Good 2001a)
- **Iterative development:** MusicXML schema definition developed alongside software to convert between MusicXML and other formats (Good 2001b)
  - Initial software: convert to/from MuseData, read from NIFF, write to MIDI (Good 2006a)
- **Usability:** balance needs of users and developers, prioritize clarity in naming conventions (Good 2006b)
- **Extensibility:** possibility for future coverage of early music and less standard notation (Good 2001a)
- **Commercial angle:** support market leaders (Finale and Sibelius) (Good 2006b)

Why XML?

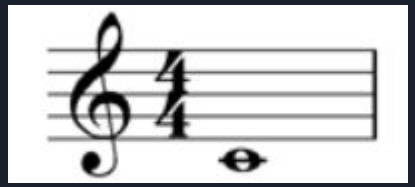
- Standardized way to represent complex, structured data
- XML programming tools widely available across languages and platforms
  - Tight coupling of formats and development tools (e.g. Finale and C/C++) was a limitation in music software development previously
  - XML → more freedom for developer to choose programming environment
- Internet-friendly
- Human-readable

(Good 2001a; Good 2001b)

Features of MusicXML

- Score file represents single movement, opus file represents multi-movement works and collections (Good 2001a)
- Does not represent presentation (e.g. pages, systems) (Good 2001a)
- Supports part-wise (measures nested within parts) or time-wise (parts nested within measures) representations of music
  - Extensible Style Sheet Transformations (XSLT) to convert between representations (Good 2001a)
- Musical data in **elements**, visual and performance information in **attributes** or special-purpose elements
  - Names of elements and attributes based on English-language musical terms (Good 2013)
  - DTD (and XSD as of 2008) defines allowable attributes and sub-elements

MusicXML Example



```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<!DOCTYPE score-partwise PUBLIC
    "-//Recordare//DTD MusicXML 3.1 Partwise//EN"
    "http://www.musicxml.org/dtds/partwise.dtd">
<score-partwise version="3.1">
  <part-list>
    <score-part id="P1">
      <part-name>Music</part-name>
    </score-part>
  </part-list>
  <part id="P1">
    <measure number="1">
      <attributes>
        <divisions>1</divisions>
        <key>
          <fifths>0</fifths>
        </key>
        <time>
          <beats>4</beats>
          <beat-type>4</beat-type>
        </time>
        <clef>
          <sign>G</sign>
          <line>2</line>
        </clef>
      </attributes>
      <note>
        <pitch>
          <step>C</step>
          <octave>4</octave>
        </pitch>
        <duration>4</duration>
        <type>whole</type>
      </note>
    </measure>
  </part>
</score-partwise>
```

Measure attributes: key signature, time signature, clef

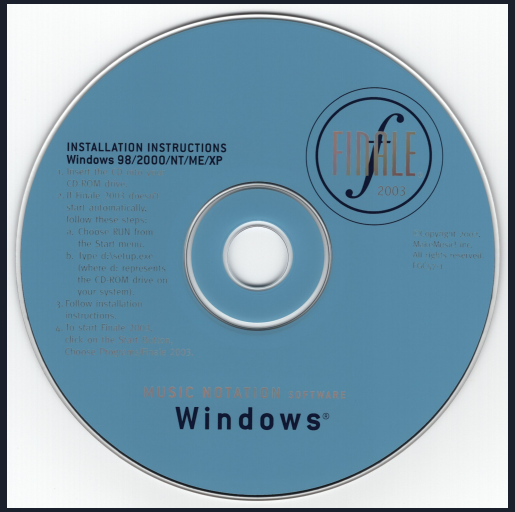
Note information: pitch, duration, type

https://www.musicxml.com/tutorial/hello-world/

Development Milestones: The First Decade

- **2000:** Recordare founded, MusicXML Version 0.1 introduced at ISMIR 2000
- **2001:** SharpEye Music Reader first commercial program to support MusicXML
- **2002:** Finale plug-in for MusicXML import/export released
- **2003:** Sibelius plug-in for MusicXML export released
- **2004:** MusicXML Version 1.0 released
- **2005:** Sibelius 4 reads MusicXML 1.1 directly from file menu
- **2007:** MusicXML Version 2.0 released
- **2008:** XSD version of MusicXML 2.0 released

(Good 2013)



https://ia803109.us.archive.org/35/items/finale2003version2003.r2windows2002/CD%20Front.jpg

Windows support for MusicXML in Finale 2003; Mac OS X support in Finale 2006; Mac Universal Binary support in Finale 2007

Development Milestones: The Next Decade

- **August 2011:** MusicXML Version 3.0 released
- **November 2011:** MakeMusic (publisher of Finale software) takes over selected assets of Recordare, including MusicXML
- **July 2015:** Further development of MusicXML moved to W3C Music Notation Community Group
- **September 2017:** MusicXML website lists 225 software applications that support MusicXML
- **December 2017:** MusicXML Version 3.1 released

(British Library 2018)

Adoption as a Preservation Format

- Library of Congress *Recommended Formats Statement*
  - XML-based formats preferred for Digital Musical Compositions (score-based representations)
  - MusicXML, MEI, and other “widely-used and publicly documented musical notation DTDs/schemas” (Library of Congress 2019)
- British Library Preservation Assessment, 2018
  - Digital Preservation Team executes evidence-based assessments of file formats
  - Covers development status, adoption and usage, software support, documentation and guidance, complexity, embedded or attached content, external dependencies, legal issues, technical protection mechanisms, other preservation risks
  - Identified preservation risks: same for all forms of XML

(British Library 2018)

MNX-Common: The Future?

- W3C Music Notation Community Group project
- Still in early development stage—specifications not ready for implementation
- Open standard for machine-readable Common Western Musical Notation

"We see MNX-Common as the next generation of MusicXML, enabling new uses that MusicXML didn't set out to support."[1]

[1] https://w3c.github.io/mnx/

References

[British Library 2018] British Library Digital Preservation Team. 2018. “MusicXML Format Preservation Assessment.” https://wiki.dpconline.org/images/2/24/MusicXML_Format_Assessment_v1.pdf

[Castan et al. 2001] Castan, Gerd, Michael Good, and Perry Roland. 2001. “Extensible Markup Language (XML) for Music Applications: An Introduction.” In *The Virtual Score: Representation, Retrieval, Restoration*, edited by Walter B. Hewlett and Eleanor Selfridge-Field, 95–102. Cambridge, MA: MIT Press.

[Good 2001a] Good, Michael. 2001. “MusicXML for Notation and Analysis.” In *The Virtual Score: Representation, Retrieval, Restoration*, edited by Walter B. Hewlett and Eleanor Selfridge-Field, 113–24. Cambridge, MA: MIT Press.

[Good 2001b] Good, Michael. 2001. “MusicXML: An Internet-Friendly Format for Sheet Music.” In *Proceedings of XML 2001*.

[Good 2006a] Good, Michael. 2006. “MusicXML in Commercial Applications.” In *Music Analysis East and West*, edited by Walter B. Hewlett and Eleanor Selfridge-Field, 9–20, Cambridge, MA: MIT Press.

[Good 2006b] Good, Michael. 2006. “Lessons from the Adoption of MusicXML as an Interchange Standard.” In *Proceedings of XML 2006*, pp. 5–7.

[Good 2013] Good, Michael. 2013. "MusicXML: The First Decade." In *Structuring Music through Markup Language: Designs and Architectures*, edited by Jacques Steyn, 187–92. IGI Global.

[Library of Congress 2019] Library of Congress. 2019. “Library of Congress Recommended Formats Statement 2019-2020.” https://www.loc.gov/preservation/resources/rfs/RFS%202019-2020.pdf