W3C Music Notation Community Group Meeting

Musikmesse Frankfurt
8 April 2016

Agenda

- SMuFL 1.2 update
- MusicXML 3.1 update
- Stories, requirements, and technical considerations
- Reception at 16:30 sponsored by Newzik

SMuFL 1.2 update

W3C Music Notation Community Meeting
2016-04-08

☐ ① **Add symbol for three-quarters flat often used in Arabic notation** `v1.2`   💬 2
#44 opened 25 days ago by dspreadbury

☐ ① **Add colon and hyphen separator glyphs for combined dynamics (e.g. f-p or mf:pp) to Dynamics range** `v1.2`   💬 0
#43 opened on Feb 24 by dspreadbury

☐ ① **Add left and right parentheses and brackets for hairpins to Dynamics range** `v1.2`   💬 0
#42 opened on Feb 22 by dspreadbury

☐ ① **Convert final SMuFL 1.2 release to W3C FSA License** `v1.2`   💬 0
#38 opened on Dec 10, 2015 by dspreadbury

☐ ① **Define cut-outs for notehead characters** `in progress` `v1.2`   💬 3
#36 opened on Nov 13, 2015 by dspreadbury

☐ ① **Add new "soft accent" character to 'Articulations' range** `v1.2`   💬 7
#35 opened on Oct 26, 2015 by dspreadbury

☐ ① **Add new 'Fingering' range** `v1.2`   💬 0
#34 opened on Oct 19, 2015 by dspreadbury

☐ ① **Add Couperin arpeggio up/down swash to 'Multi-segment lines' range** `v1.2`   💬 0
#33 opened on Oct 19, 2015 by dspreadbury

SMuFL 1.2 status

– Currently 30 issues in scope

– 23 are (largely uncontroversial) proposals for new characters and/or ranges

– 3 (#7, #8, #9) recommend changes in form to existing characters

– 2 (#20, #28) are proposals for stylistic alternates for existing characters

– 1 (#36) is a proposal for extending metadata for cut-outs to include noteheads

– 1 (#38) is some housekeeping to convert to the appropriate W3C license

– Expect to complete this work no later than end of Q3 2016

MusicXML 3.1 Update

W3C Music Notation Community Meeting
2016-04-08

MusicXML 3.1 Overview

- An incremental update to maintain application compatibility while we work on longer-range issues
- Focused on
  - Support for SMuFL font symbols
  - Documentation and other bug fixes
  - Small set of additional new features and updates
- Expect to complete no later than end of Q3 2016

Author ▾   Labels ▾   Milestones ▾   Assignee ▾   Sort ▾

☐ ① **Clarify key-octave documentation/definition** `Documentation` `V3.1`   💬 0
  #126 opened 11 days ago by ebrooks01   ⏱ updated just now

☐ ① **Clarify definition of String element in MusicXML** `Documentation` `V3.1`   💬 0
  #127 opened 7 days ago by ebrooks01   ⏱ updated 23 seconds ago

☐ ① **Add size / scaling information for images** `In Progress` `Reflow` `V3.1`   💬 1
  #3 opened on Aug 13, 2015 by mdgood   ⏱ updated 8 days ago

☐ ① **Allow specification of SMuFL unique glyph names in other- elements** `In Progress` `SMuFL` `Symbols` `V3.1`   💬 1
  #107 opened on Nov 6, 2015 by mdgood   ⏱ updated 8 days ago

☐ ① **Allow specification of SMuFL unique glyph names for accidentals** `In Progress` `SMuFL` `Symbols` `V3.1`   💬 1
  #109 opened on Nov 6, 2015 by mdgood   ⏱ updated 8 days ago

☐ ① **Add staff division symbols** `In Progress` `SMuFL` `Symbols` `V3.1`   💬 1
  #11 opened on Aug 13, 2015 by mdgood   ⏱ updated 10 days ago

☐ ① **Convert final MusicXML 3.1 release to W3C FSA License** `Documentation` `V3.1`   💬 0
  #114 opened on Dec 10, 2015 by mdgood   ⏱ updated on Dec 10, 2015

☐ ① **Add complete SMuFL lyrics range** `Appearance` `SMuFL` `V3.1`   💬 2
  #101 opened on Aug 20, 2015 by mdgood   ⏱ updated on Nov 10, 2015

MusicXML 3.1 Status

- Currently 37 open issues in scope
  - SMuFL symbol support (28 open issues)
  - Documentation and other bug fixes (3 open issues + 9 closed)
  - Small set of additional new features / updates (6 open issues)
- Development plan
  - Propose an implementation as a GitHub comment
  - Discuss on GitHub
  - Pull request to merge the actual changes into the schemas
  - Please respond soon to proposals; we want to move quickly

Stories, Requirements, and Technical Considerations

W3C Music Notation Community Meeting
2016-04-08

Topics

- User Stories
- Document Content Types - common Western music notation + ???
- Conflicts in Capture, Representation and Rendering
- Encoding Profiles
- Requirements Matrix
- Architectural Demo: Document-Based Development
- Architectural Recommendations

User Stories

- We now have 70+ User Stories (narrative use cases)
- Many evolved from shorthand descriptions into short narratives
- Multiple contributors
- How is this process working?
- How complete is it?

Document Content Types

- No constraints on semantics => semantic interpretation is hard
- Common Music Notation (whatever *that* means!) is a core type
- Guitar Tablature and Chord/Lyric Sheets have cultural currency
- Neumes and Mensural Notation imply fuzzier semantics
- Open-ended graphical notation best suited by a graphical standard, not a semantic one
- We do not need to represent every work in the compositional canon

Conflicts in Capture, Representation and Rendering

- Many stories reflect a need to capture what is "actually there" in a document
- Capturing what is "actually there" means <u>omitting what is not there</u>
- Conclusion: faithful capture implies documents that are semantically...
  - inconsistent
  - incomplete
  - ambiguous
- Many other stories reflect a need to accurately and flexibly render notation
- Accurate rendering implies semantic consistency, completeness, specificity
- Some rendering stories are based on encoding a score's "original geometry"; others must dynamically cook up a fresh layout.
- How do we reconcile these conflicts?
- *Do* we reconcile these conflicts?

Encoding Profiles

- An *encoding profile* is a set of guarantees made by documents that conform to it
- Profiles reflect "encoding intent"
- A profile is a set of constraints, not just a subset of the schema
- Standardized profiles can solve many portability problems
- Corollary: *ad hoc* profiles create many portability problems :)
- "Loose" profiles are appropriate for archival capture
- "Tight" profiles are appropriate for rendering and semantic manipulation
- New features can be soft-launched by not adding to profiles

Requirements Matrix

- Initial attempt to derive and categorize requirements
- Traceable back to user stories
- About 50 requirements so far
- Many require a great deal of elucidation and further work

Architectural Live Demo:

Discover Your Roots
(Notation DOM example)

Architectural Recommendations

- Define/Adopt Encoding Profiles
  - Cake: it's good to eat <u>and</u> good to have!
- Gear Encoding towards a DOM Approach
  - We need much more than an interchange and archival format
- Standardize Key Aspects of Layout and Performance Rendering
  - Give content creators and publishers meaningful, yet optional control
- Use CSS for Visual and Performance Facets

  - Styles run the gamut from general to specific
  - Different styles are appropriate to different contexts
  - CSS is a best-in-class answer to these needs