Week 11
Music as Data

Roger B. Dannenberg
Professor of Computer Science and Art & Music
Carnegie Mellon University

MIDI Files as Music Representation

- MIDI messages are limited to performance information
- Standard MIDI Files are somewhere between performance and symbolic notation, providing:
  - Time Signature
  - Tempo
  - Key Signature
- If timing is quantized to beats, you can recover a lot of notation:
  - Bar lines
  - Time signatures
  - Tempo and tempo changes
  - Keys and key changes

Limitations of Standard MIDI Files

- Parameters outside of MIDI:
  - Think of OSC data types and name space
  - Per-note parameters (MIDI control change messages affect entire channel)
- Accidentals (A ♭ or G♯ ?)
- Staves and Voices
  - But every track can have name
- Beams, Slurs, Clef
- Articulation Markings, Dynamics
  - Essentially all annotations
- Graphics
  - Stem direction
  - Spacing
- MIDI *does* have: tempo, time signature, key signature

Note Lists Are the "standard" computer music representation

- Flat list of "notes" with time and duration
- Each note specified by parameters:

```
0.0 0.5 i1 60 1.0 0.3 0.5 0.1  6.2
0.0 0.5 i1 65 1.0 0.3 0.6 0.15 6.2
0.5 0.4 i1 67 0.2 0.1 0.3 0.1  4.0
1.0 2.0 i2 78 0.5 0.4 1.0 0.4  6.0
...
```

Adagio Is a Simple Text-Based Music Representation

- Machine-readable, but sensible to humans
- Performance/Synthesis data only
- (Somewhat) Extensible
- Examples:

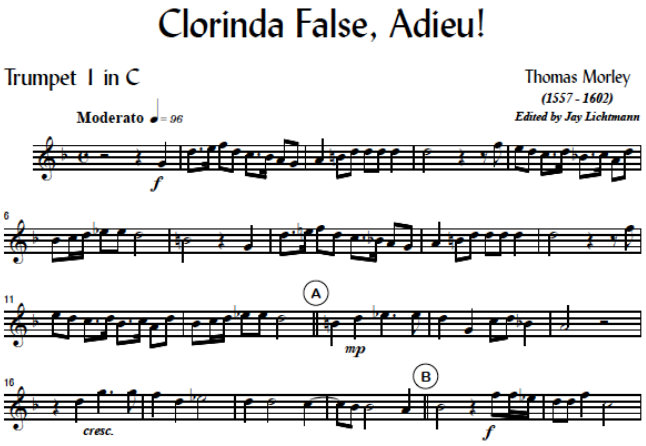
```
!tempo 100
t0 c4 q nq
cs4 q. Lff V3
th ef w
~23(10) *control
```

Comment Character

```
!CLOCK * turn on MIDI clock
!RAMP X10 X100 Q W2 * linear ramp
!CALL trill(A5,W,2,S,Lmf) T278 V1
!SETI myvar 75
!SETV myarray 5 -4 T100
```


Adagio Example

Started with PDF – need to change key…



Adagio Example
Hand-entered Adagio from Score

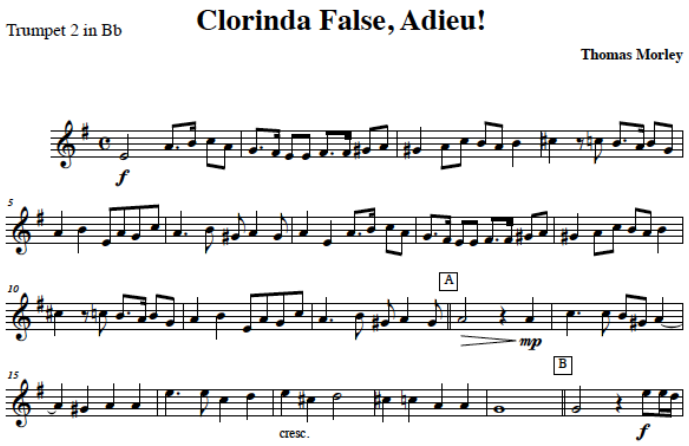
```
d h; g i.; a s; bf i; g
f i.; e s; d i; d; e i.; e s; fs i; g
fs q; g i; bf; a ; g; a q
b q; r i; bf; a i.; g s; a i; f

g q; a; d4 i; g; f; bf
g q.; a i; fs i; g q; fs i
g q; d; g i.; a s; bf i; g
f i.; e s; d i; d; e i.; e s; fs i; g
fs q; g i; bf; a; g; a q
```

```
*m10
b q; r i; bf i; a i.; g s; a i; f
g q; a; d4 i; g; f; bf
g q.; a i; fs i; g q; fs i
*letter A
g h; r q; g q
bf q.; bf i; a; fs; g h

*m15
fs q; g q; g q;
d5 q.; d i; bf q; c
d5; b; c h
…
```


Adagio Example
Adagio to SMF using Nyquist

```
;; Nyquist Program to Translate to Std. MIDI File
set seq = seq-create()
;; open input file and read Adagio file into seq
set inf = open("clorinda-false-adieu.gio", "r")
exec seq-read(seq, inf)
exec close(inf)

;; open output file and write SMF
set outf = open-binary(
    "clorinda-false-adieu.mid", direction: :output)
exec seq-write-smf(seq, outf)
exec close(outf)
```

Adagio Example
SMF to PDF using Finale



Allegro (Included in Audacity)

- More recent than Adagio
- Libraries in Serpent and C++
- Same general idea: text representation for score
- Attribute/Value syntax:

```
t0 -chani:2 -keyi:60 -gater:100 -pitchr:60 -durr:0.7
```

But thanks to some syntactic sugar, you can also write….

```
T0 V2 C4 LMF Q
```

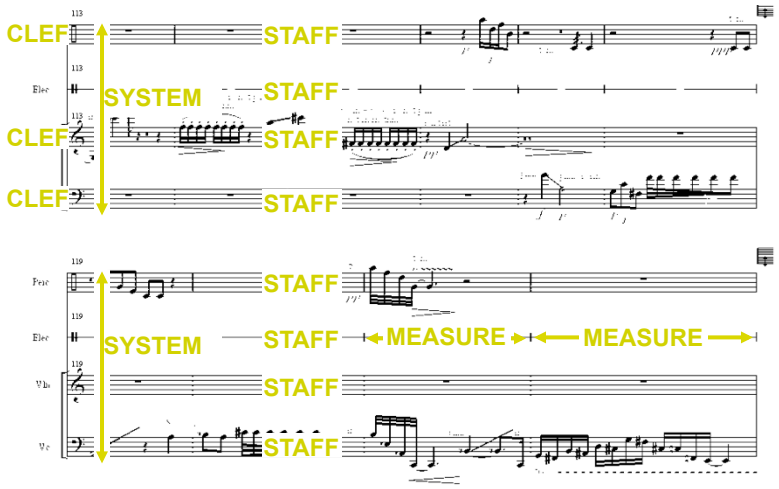
Music Notation

- Editing requires semantics
  - Placement determined by beats, which are determined by symbolic durations
  - Operations such as transposition require interpretation of pitch, key signatures, etc.
- But semantics are not formal
  - Music is not a graph on a strict coordinate system
  - Music is full of implied meanings and vague instructions

What you should know…

- The general structure of music notation
- Some examples of difficult problems
- Some systems for encoding music notation
- Limitations of MIDI for notation
- General capabilities (and lack thereof) in current music notation software systems

General Structure of Music Notation



Rules for Pitch

Clef **gives pitch reference, this clef means "G4" is here**

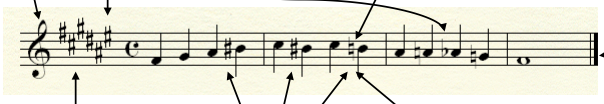
Note heads centered lines and spaces, 7 steps/octave, (white notes)

12 ***chromatic tones*** **per octave**

Sharp **= up 1 chromatic tone**

Flat **= down 1 chromatic tone**

Natural **= as written**

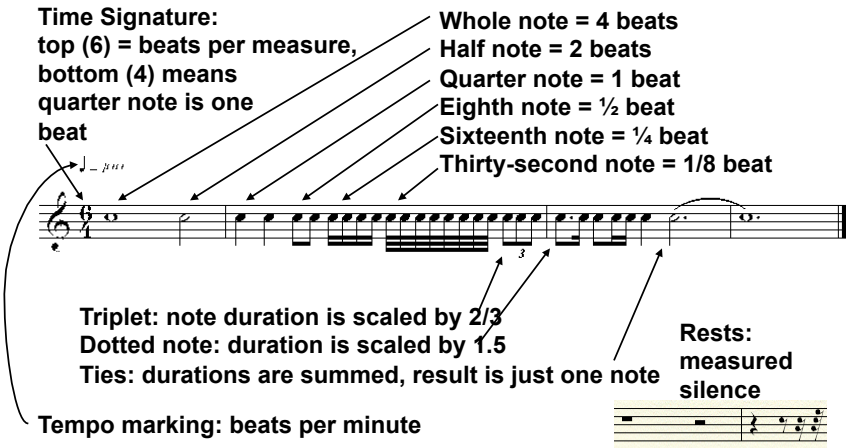


Key Signature: **default sharps or flats**

Accidental: **overrides default from key signature**

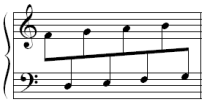
Accidental becomes default for rest of measure

Rules for Rhythm (Time)



Difficult Representation (and Editor Interface) Problems

- Scope of symbols
  - Accent marks, loudness indications, etc. can apply to
    - individual notes
    - the staff
    - the system
- Multiple tempi (in contemporary music, see 8:15 in https://www.youtube.com/watch?v=12j1wdKE4zU)
- Voices
  - A "line" of music
  - Merging and splitting – multiple voices p
  - Sometimes rests are missing or shared
  - Voices can cross staves
- Beats may not add up to full measure

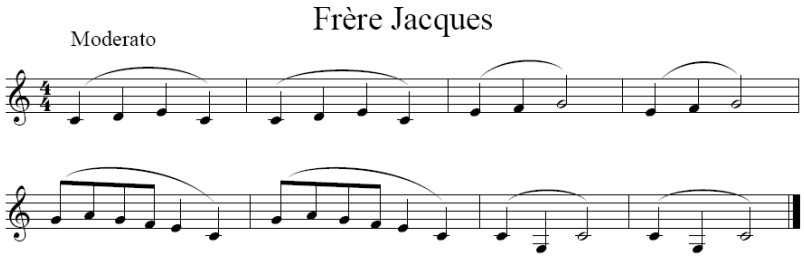


Encoding Systems: GUIDO

- Plain text, human readable, platform independent
- by Holger H. Hoos, Keith A. Hamel, Kai Renz, Jurgen Kilian
- Based on musical concepts as opposed to strictly graphical features
- Alternative strategies:
  - Binary formats (NIFF, SMF)
  - General but complex notation (DARMS, SMDL)
  - Graphics only (cmn)

Example of GUIDO

```
[ \title<"Frere Jacques">
\tempo<"Moderato"> \clef<"treble"> \meter<"4/4">
\slur(c1/4 d e c) \slur(c d e c)
\slur(e f g/2) \slur(e/4 f g/2)
\slur(g/8 a g f e/4 c) \slur( g/8 a g f e/4 c)
\slur(c g0 c1/2) \slur(c/4 g0 c1/2) ]
```

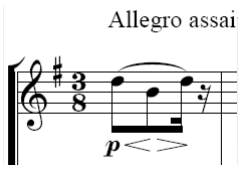


Some Details

- **`c#1*1/4.`** = dotted quarter note middle C#
  - "`*`" is a separator
- **`_*1/4`** = rest
- Sequences: **`[ … ]`** → sequential in time
- Segments: **`{ … }`** → simultaneous
- Tags:
  `\tagname<parm=value,parm=value>(…)`

Example of Advanced GUIDO

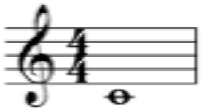
```
\tempo<"Allegro asai",dx=0,dy=9.6>
\space<7.36> \i<"p",dy=-5.76>
\beam(\slur<y=1.92>(
   \crescBegin<dx=1.28,dy=-5.76> d2/8 \space<6.4>
   \merge( b1/16 \crescEnd<dy=-5.76>
           \dimBegin<dx=1.28,dy=-5.76> b )
   \space<6.4> d2/16 \dimEnd<dx=0,dy=-5.76>)) _/16…
```



Encoding Systems: MusicXML

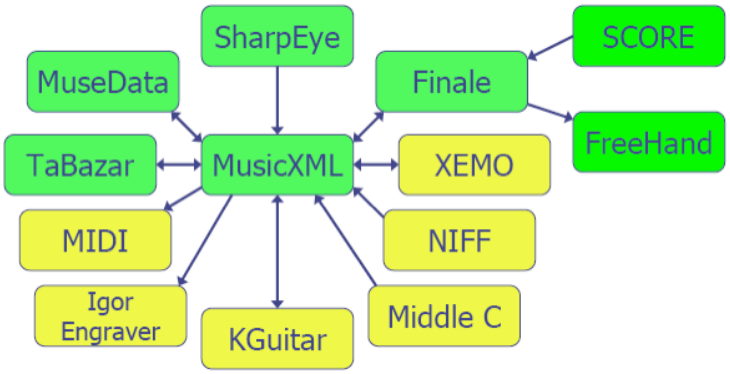
- Michael Good, Recordare LLC, now part of Make Music, Inc.
- XML-based representation for music notation
- Supports notation, analysis, information retrieval, and performance applications
- Used by Finale & SharpEye Music Reader, Sibelius, MuseScore, >100 more

Example



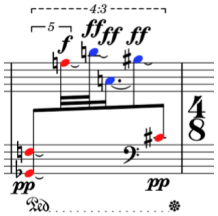
```
<?xml version="1.0"
    encoding="UTF-8"
    standalone="no"?>
<!DOCTYPE score-partwise PUBLIC
  "-//Recordare//DTD MusicXML…//EN"
  "http://www.musicxml.org/…">
<score-partwise>
 <part-list>
  <score-part id="P1">
   <part-name>Music</part-name>
   </score-part>
 </part-list>
 <part id="P1">
  <measure number="1">
   <attributes>
    <divisions>1</divisions>
    <key>
     <fifths>0</fifths>
    </key>
   <attributes>
    <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>
```

MusicXML for Interoperability



Lillypond

- Designed for high-quality music typesetting
- 100K lines of C++ and Scheme
- Active following
- [ ] for beams, ( ) for slurs
- Helmholz notation: C, C c c' c'' …
- Numerals for duration (1 2 4 …)
- Tuples: a2 \times 2/3 { b4 b b }
- Latex-like keywords: \key

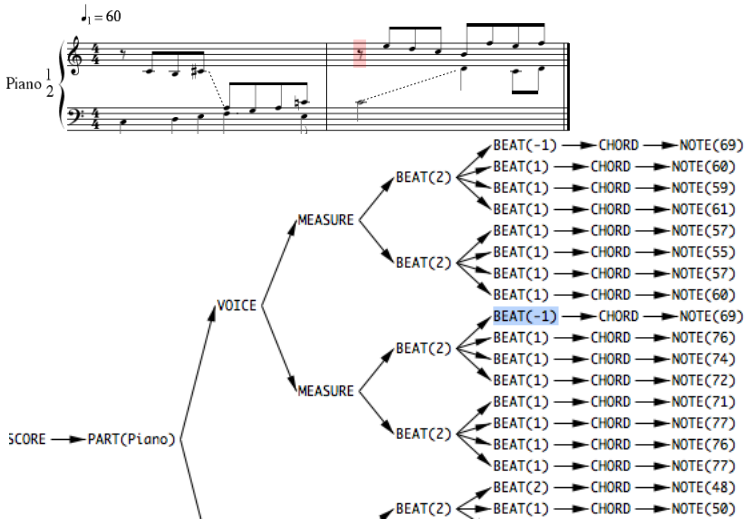


Example from Wikipedia

Lillypond example

```
\version "2.14.1"
\include "english.ly"

\score {
  \new Staff {
    \key d \major \numericTimeSignature
    \time 2/4
    <cs' d'' b''>16 <cs' d'' b''>8.
    %% Here: the tie on the D's looks funny

       %% Too tall? Left-hand endpoint is not aligned with the B tie?
    ~
    <cs' d'' b''>8 [ <b d'' a''> ]
  }
}
```


ENP - Expressive Notation Package



Current Music Notation Systems

- Nearly all published music is now done with computers
- At least two powerful commercial systems: Finale, Sibelius
- Some open source editors: MuseScore, LillyPond
- A web-based (in Flash) editor: NoteFlight
- Combine semantic-based layout with open-ended manual layout
- Lacking:
  - Constraint systems to maintain manual placement when automatic spacing changes
  - Tracking change from score to parts and vice-versa (at least this seems overlooked or limited)
  - Robust interchange formats – but MusicXML offers a path
  - APIs giving programmable access to notation (but this exists in at least limited forms for more than one notation system)

Operations On Scores

- Reference: Laurie Spiegel, "Manipulations of Musical Patterns." In *Symposium on Small Computers in the Arts*. Los Angeles: IEEE Computer Society. 1981. pp. 19-22.
- Transposition: add constant offset to pitch (expressed in log frequency units)
  - can apply to other parameters, especially amplitude (in dB)
- Reversal
  - Pitch reversal: $p_i' = C - p_i$ (a.k.a. *inversion*)
  - Order reversal: $n_i' = n_{N+1-i}$ (i=1…N) (*retrograde*)
  - Reversal of structure: abc_qrs_xyz → xyz_qrs_abc

Operations on Scores (2)

- Rotation:
  - Sequences: $n_i' = n_{((i+C) \bmod N)+1}$ (i=1…N)
  - Chord inversion is a form of rotation
- Phase Offset and Phasing
  - Rotation relative to a (same or different) cyclic pattern
  - Cyclic patterns of unequal length
  - Parametric phasing as in isorhythmic motet, e.g.:
    - Pitch sequence of length 9, 4 times
    - Rhythmic sequence of length 12, 3 times

Operations on Scores (3)

- Rescaling
  - Augmentation/diminution (scaling the number of beats in each note)
  - Tempo change (scaling duration of a beat)
  - Reversal is related to a special case (-1)
- Interpolation
  - Linear (or other) interpolation between points
  - Melodic “interpolation” over chords
  - Variations of a theme
  - Embellishments with trills and ornaments

Operations on Scores (4)

- Extrapolation (no operational definition)
- Fragmentation: Isolation and reuse of a sub-pattern
  - Generally along time axis
  - Can be through separation and application of different parameters
- Substitution: of elements or sub-patterns
  - Without rule
  - By some orderly process
  - Individually or as part of a coordinated exchange
  - Perception of substitution requires perception of pattern

Operations on Scores (5)

- Combination: mixing, overdubbing, counterpart, harmony,…
- Sequencing: append, splice, delete, …
  - Temporal dimension of *Combination* (above)
  - Disjunct/conjunct/overlapped
  - Continuous/discrete
- Repetition
  - Fundamental to music
  - Repetition in time
  - Duplication of voices or sound sources

Hierarchical Representations

- You've seen hierarchy already in MusicXML and GUIDO
- Combining flat note-list representation with combination/transformation primitives gives a hierarchical representation
- This is the basis for many functional-style programming languages, including Nyquist

Nyquist can be viewed as a Hierarchical Score Representation

- Combination:
  - SIM – simultaneous
  - SEQ – sequential
- Transformation:
  - TRANSPOSE – pitch offset
  - LOUD – loudness/dynamics offset
  - SUSTAIN – "articulation"/duration scaling
  - STRETCH – time stretch/tempo change
  - AT – shift
- Structure:
  - DEFUN – introduce named phrase
  - SEQREP, SIMREP – iteration

Nyquist Example

```
(defun mel1 ()
  (seq (note c4 q)
       (note d4 q)
       (note e4 h)))
(defun mel2 ()
  (seq (note c5 i)
       (note b4 i)
       (note a4 i)
       (note b4 i)
       (note g4 q)))
; sequencing
(seq (mel1) (mel2))
; dynamics
(seq (mel1)
     (loud -90 (mel2)))
```

```
; loop and tempo
(seqrep (i 4)
 (stretch 0.25
  (seq (mel1) (mel2))))
; transpose
(defun mel ()
  (seq (mel1) (mel2)))
(seq (mel)
     (transpose 2 (mel))
     (mel))
; articulation/legato
(seq (mel)
     (sustain 0.5 (mel))
     (sustain 1.5 (mel)))
; sim and at
(sim (mel) (at 0.1 (mel)))
```


Hierarchy in Scores Appear In Many Places/Perspectives

- Score
  - Page
    - System
      - Staff
        - Voice
- Beams
- Phrases
  - Slurs
- Score
  - Sections
    - Measures
      - Beats

Multiple Hierarchies (aka Heterarchies) are important in music and complicate representations

- Hierarchies can be independent
  - Measures cut across voices
  - Slurs can cross bar lines
- Even hierarchies of a given type can overlap:
  - Notes can participate in the end of one phrase and the beginning of the next
  - Sections can begin/end in the middle of measures
  - Voices can cross staves
- Note that this does not lead to a clean, human-readable, linear text encoding

Conclusions

- Music defies a single representation
- Compare to Mathematics and Math notation
- Notation (♪♫) is easy for musicians to read, but difficult to manage in general
- Many composers work with simpler representations:
  - Note lists (abstract parameter data)
  - Midi files (performance data, not notation)
  - Simple music notation software for visualization
  - Full-blown notation systems to prepare scores for performance by humans