A crash course in the history of music modelling

Frans Wiering
f.wiering@uu.nl

Department of Information and Computing Sciences

Universiteit Utrecht

Abstract

A computer is a symbol manipulation machine. In order to be able to process music automatically, it has to be captured in symbols the computer understands. The output of the manipulation in turn needs to be interpretable as a meaningful statement in the music domain. There is thus a high-risk, twofold translation process going on, where modelling decisions may lead to revealing insights as well as to perplexity. Moreover, the affordance of models—their potential to be manipulated—may influence how we see music, consciously or unconsciously. During the nearly 70 years that music has been studied computationally it has been modelled in many different ways. This talk gives a brief overview of this development, highlighting properties and limitations of some of the most popular approaches.

Contents

- first, a disclaimer…
- case study from my own past
  - 1991, nicely in the middle of the history
  - DARMS encoding in action
  - several fundamental issues
- generalised model of music processing
- 7 core music processing tasks
  - for each, one (old) encoding system
  - modelling of pitch
- differences and interoperability
- the perspective of the researcher
- data creation and availability

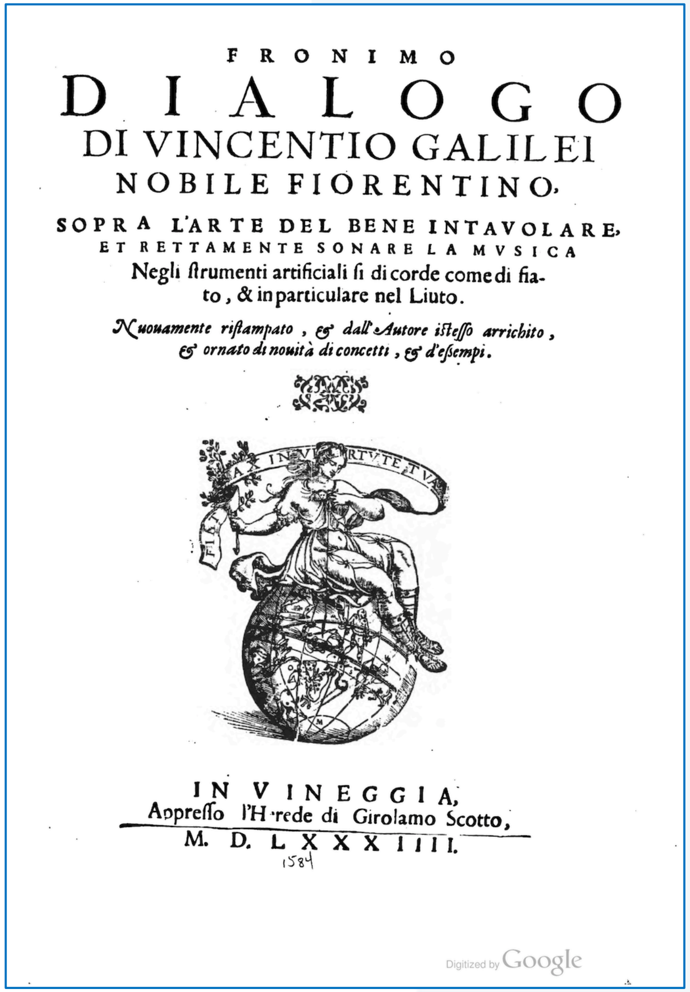
musicXML™

music21

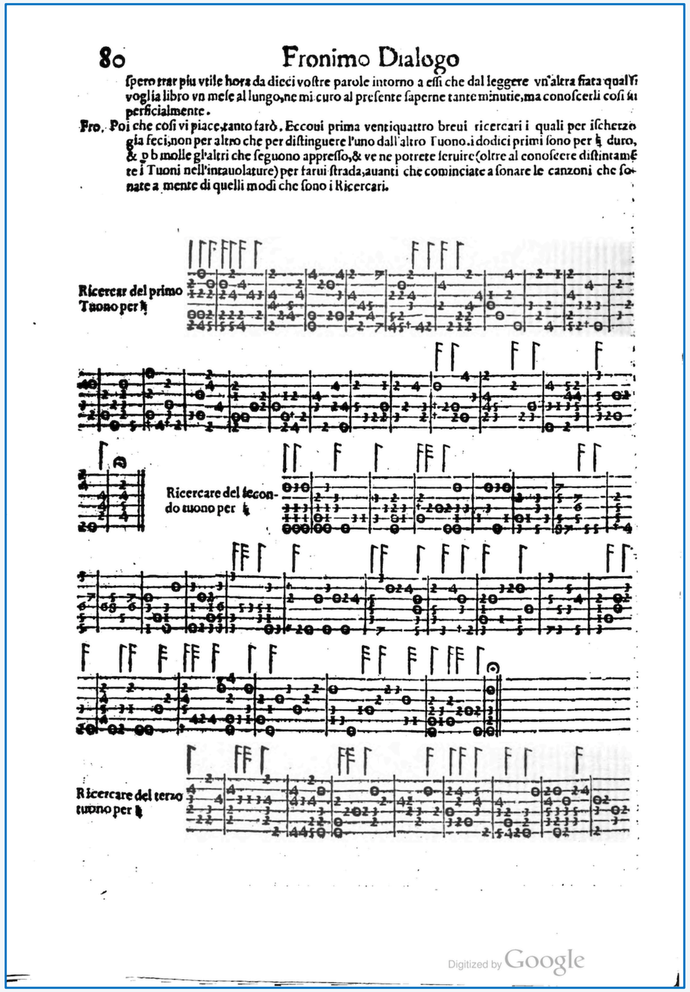
MEI

not addressed in this talk

1991: interesting, difficult source


FRONIMO
DIALOGO
DI VINCENTIO GALILEI
NOBILE FIORENTINO.
SOPRA L'ARTE DEL BENE INTAVOLARE,
ET RETTAMENTE SONARE LA MVSICA
Negli ſtrumenti artificiali ſi di corde come di fia-
to, & in particulare nel Liuto.
Nuouamente riſtampato, & dall'Autore isteſſo arrichito,
& ornato di nouità di concetti, & d'eſempi.

IN VINEGGIA,
Appreſſo l'Herede di Girolamo Scotto,
M. D. LXXXIIII.
1584



ſpero trar piu vtile hora da dieci voſtre parole intorno a eſſi che dal leggere vn'altra fiata qual ſi voglia libro vn meſe al lungo, ne mi curo al preſente ſaperne tante minutie, ma conoſcerli coſi ſuperficialmente.

Fro. Poi che coſi vi piace, tanto farò. Eccoui prima ventiquattro breui ricercari i quali per iſcherzo gia feci, non per altro che per diſtinguere l'uno dall'altro Tuono. i dodici primi ſono per ♮ duro, & p b molle gl'altri che ſeguono appreſſo, & ve ne potrete ſeruire (oltre al conoſcere diſtintamente i Tuoni nell'intauolature) per farui ſtrada, auanti che cominciate a ſonare le canzoni che ſonate a mente di quelli modi che ſono i Ricercari.

The usual manual approach

fret	0	1	2	3	4	5
course						
1	G	G#	A	Bb	B	c
2	c	c#	d	eb	e	f
3	f	f#	g	g#	a	bb
4	a	bb	b	c'	c#'	d'
5	d'	eb'	e	f	f#'	g'
6	g'	g#'	a'	bb'	b	c''

24 pieces

- time-consuming
- tedious
- error-prone

➢ automate?

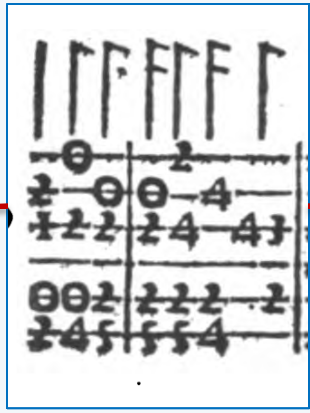
Workflow

```
20-12Q 40-2-0E 52-20E. /
52-20S 52-4-2E 42--4S ---4 -2-3E /
```

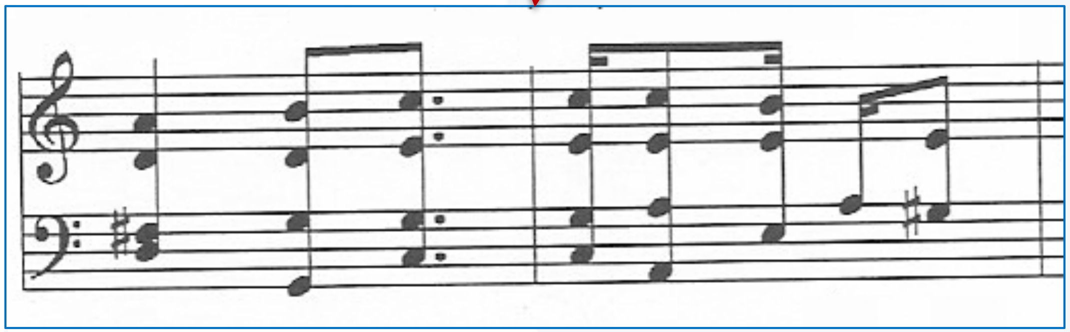
```
!I1
!T#6%200,!G,!F
!N0 !U
!/11|-50
12Q,0@ 2$,2@ 0$,6@ 1$,8@ 2$
12E,0@ 4$,2@ 0$,6@ 2$,10@ 0$
12E.,0@ 5$,2@ 2$,6@ 2$,8@ 0$
/

...
!-50

54Q,50Q,7#Q,5Q
55E,50E,8E,1E
56E.,51E.,8E.,4E.
/

...
```

1. encode tablature in DARMS-like format
2. transcribe to regular DARMS code
3. render with Note Processor software
4. study!

Partial solution



what we got



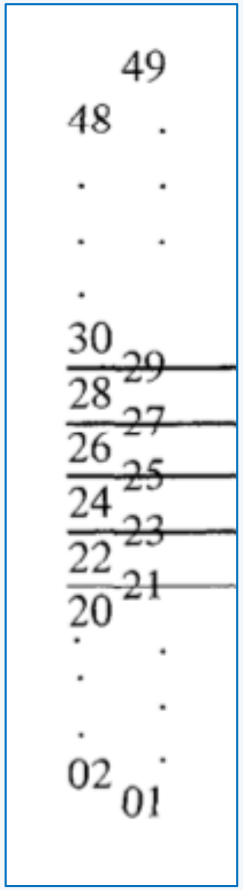
what we hoped for

- we get pitches and onsets
- we don’t get voice leading, durations
  - in fact, still a research topic (De Valk 2015)
- we lose all information about courses and frets

Indirect pitch representation

- tablature encodes finger position
  - indirectly chromatic
  - relative pitch
- DARMS encodes vertical position on staff (vp)
  - indirectly diatonic
  - clefs ‘anchor’ vps to note names

- to be inferenced/decided
  - tuning, absolute pitch
  - diatonic reading



vp encoding
(Erickson 1975)

Conversion in detail

integer representations after Brinkman (1986)

- cpc: continuous pitch class
- pc: pitch class
- nc: name class
- oct: octave

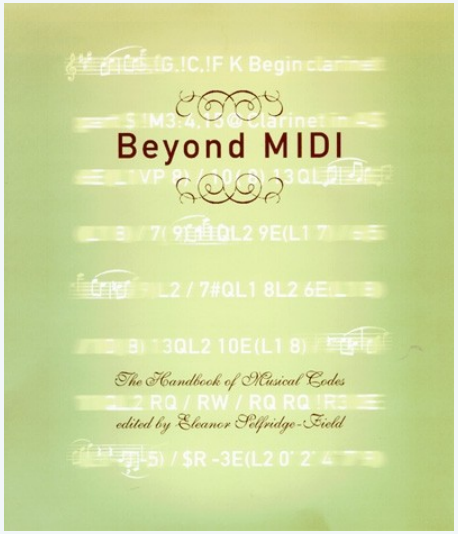
DARMS

- vp: vertical position
- acc: accidental

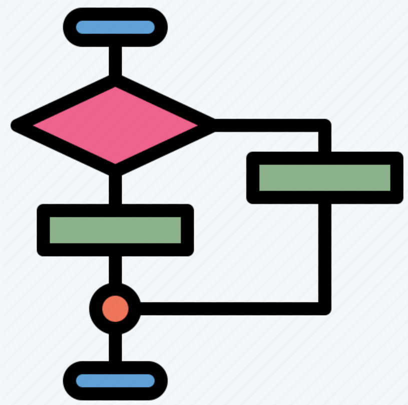
Step	Representation	Example
	Lute code	--1---
tuning, fret lookup calculate	cpc	42
calculate	<oct,pc>	<3,6>
nc lookup	<oct, pc, nc>	<3,6,3>
calculate context check	<vp, acc>	<7,#>
merge in DARMS token	DARMS code	7#Q

What this case shows

- illustrates then prevailing DIY approach
  - create own encoding and software
- music encoding dilemma
  - musical logic or graphical representation?
- encoding vs. internal representation
  - alphanumeric encoding
  - integers, lists, tuples
- internal operations entirely in terms of arithmetic, table lookup, symbol manipulation
  - there is *nothing intrinsically musical* going on inside the computer

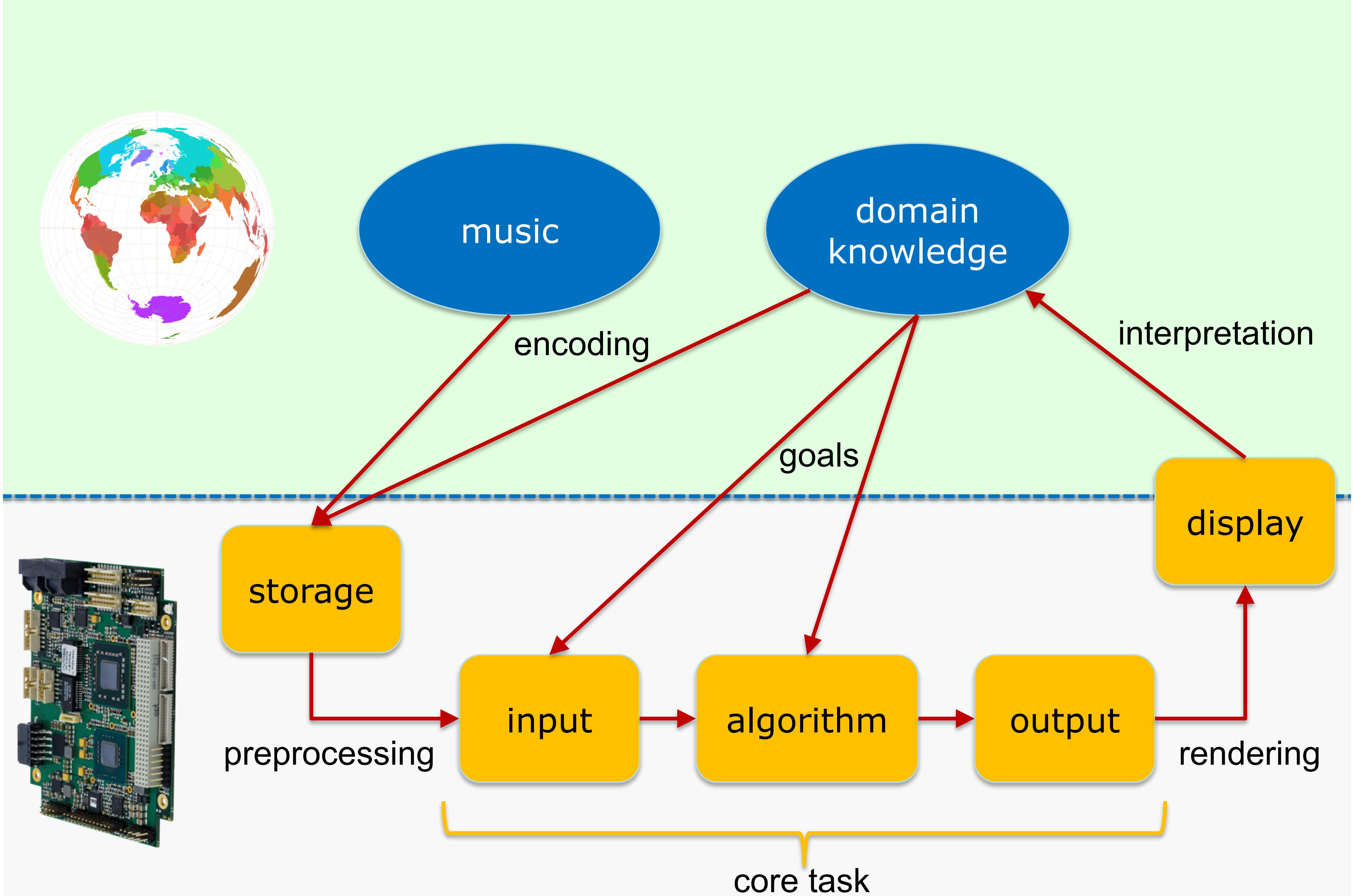


Selfridge-Field 1997 describes c. 80 encodings



it's an algorithmic process

case study suggests a general workflow for algorithmic music processing


music
domain knowledge
encoding
interpretation
goals
storage
display
input
algorithm
output
preprocessing
rendering
core task

What are those core tasks?

1. generation
   - composition
   - theory testing
   - missing data
2. preservation
   - source digitisation
   - conservation of heritage
3. rendering
   - score generation
   - performance
4. analysis
   - formalised / quantitative methods
   - corpus-based research
5. transformation
   - change notation type
   - adapt for performance
6. exploration
   - find patterns and regularities
   - select for use
7. relating
   - metadata
   - versions
   - knowledge

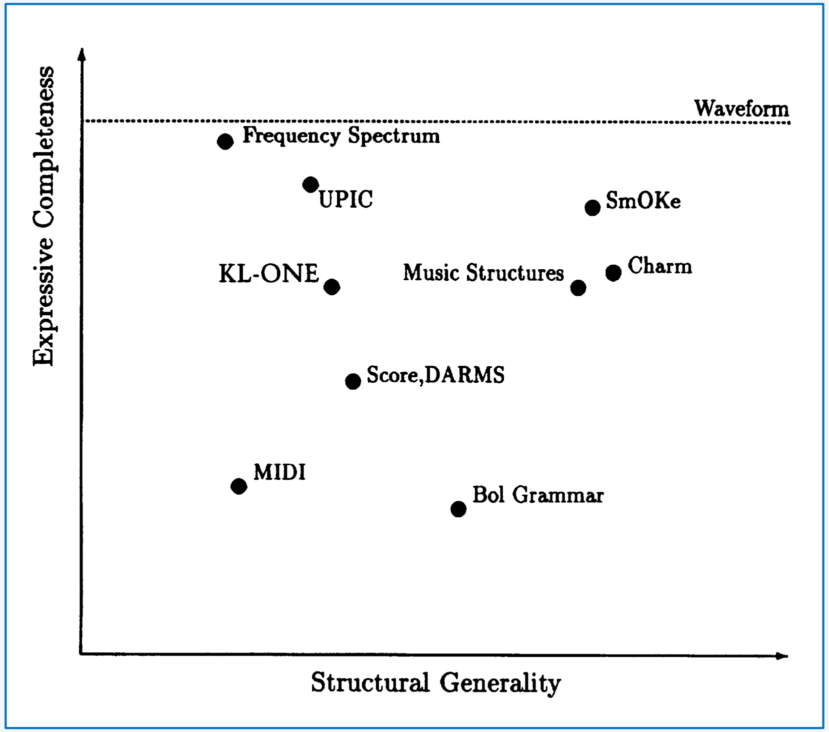
preliminary high-level typology

Encoding has a special place

- peripheral yet often necessary task (even today)
- creation/choice of encoding system influenced by
  - intended core tasks
  - (manual) data entry process
  - inclination towards minimising preprocessing
  - …other factors…
- creation of new systems is now less common
  - other input modes than direct encoding
    - OMR, GUIs, audio analysis
  - availability of significant corpora
  - increasingly refined interoperability

Representation design

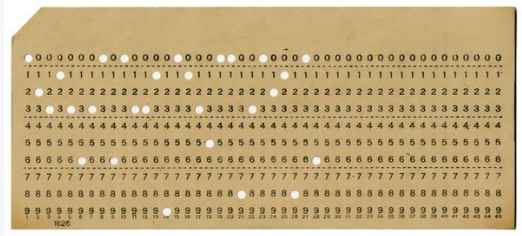
- David Huron (1992)
  - ‘it is not possible for a signifier to represent all properties of a signified’ (p. 9)
  - ‘many of the most useful forms of representation achieve their power by virtue of being able to dispense with detail deemed irrelevant’ (p. 10)



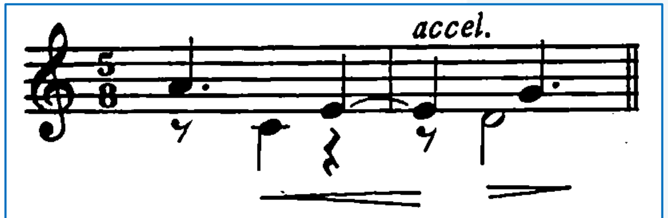
- Wiggins et al. (1993)
  - tradeoff of structural generality and expressive completeness
  - ‘the problem for the would-be constructor of a general purpose system of notation—one simply cannot anticipate all the purposes to which it may be put’ (p. 41)

Showcase

Music III: generation



- ‘a program for synthesizing music and psychological stimuli’ (Mathews 1961), written in FORTRAN
- two parts: ‘orchestra’ and ‘score’
- score consists of instructions (OP Code) with parameters



- marked in red:
  - P1: instrument
  - P2 duration
  - P5 pitch (oct.pc)

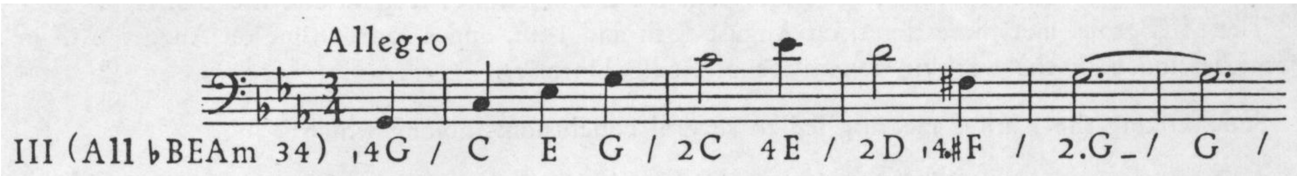
- pitch frequencies specified in orchestra

TABLE II — EXAMPLE OF A COMPOSITION

OP Code	P1	P2	P3	P4	P5	Comments (numbers refer to comments in text)
GEN	10	1				1
GEN	11	2				1
TME			1000			2
RST	2	1				3
	1	3	50		3.9	4
	1	2	55		3.4	4
	2	2	53		3.0	4
MES						5
TME			750			
RST	2	1				6
		4	60		3.2	6
	1	2			3.4	
	1	3			3.7	
MES						
TER						7

Plaine And Easie Code: preservation

- ‘an accurate shorthand for music notation, especially useful for incipits and excerpts’ (Brooks et al. 1964)
- meant for bibligraphic projects, RISM in particular
- monophonic, not suited for complete works



- 2019: ,,G/,CEG/24'CE/D,xF/2.G+/G/ (RISM OPAC)
- pitch representation: oct, nc, acc
  - between graphical and logical representatio

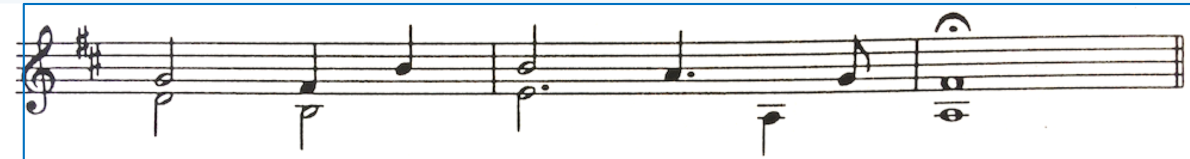
DARMS: rendering

- created by Stefan Bauer-Mengelberg et al. from 1963
- for high-quality engraving, graphically oriented
- how to render 2-D polyphony in 1-D encoding?
  - instrument numbers
  - linear decomposition
- inputting is optimised
- (non-trivial) 2-D coordination left to the rendering engine

```
IØ !M4:4,ØØ@Allegro,|QU| = 11Ø$
I1 !G RQ RE 9E_<,VF 9#Q._ 7#E_ /
I2 !G 1HJ,VF E 1-E_ 2QJ 7
I3 !C RW / RW /
I4 !F RQ 17H_,VF 2Q_ /
I1 (8_ 7*L 6) (3L 1-) REQ /
I2 (2 1-_) 2Ø-QL 18-E REQ /
I4 7#Q 27!C 8Q 33E REQ /
```

Instrument codes (Brinkman 1986)

complex issue in all score encoding systems



```
!G !K2# !& !U 3H 2Q 5 / H 4Q. 3E / 2W; //
        & !D 0H -2   / 1. -3Q   / W   // $&
```

Linear decomposition (Dydo 1991)

HUMDRUM: analysis

- David Huron (1988)
- analytical toolkit + *protocol* for representation
- structural isomorphism: structural relations between elements are preserved
- 2-D structure
  - sequential event in spines
  - concurrent events in lines
- spines may contain notation, text, analytical information,…
- tools may generate new spines, e.g. harmonic analysis

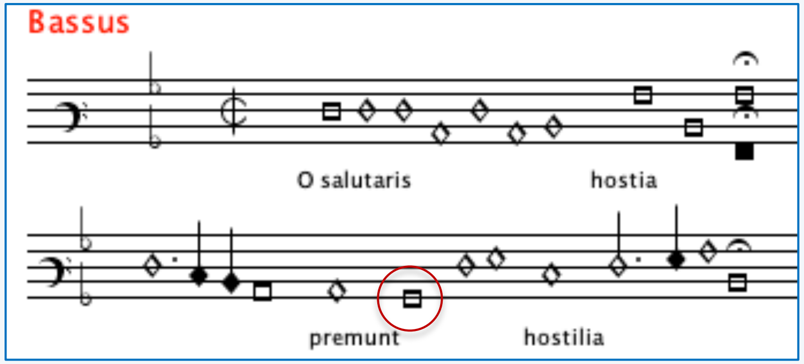

```
**beat  **kern     **silbe  **btstart  **mtstart  **rtstart
*       *clefG2    *        *          *          *u=msec
*       *k[b-e-]   *        *          *          *MM60
*M4/4   *M4/4      *M4/4    *          *          *
*MM60   *MM60      *        *          *          *
=1      =1         =1       =1         =1         =1
1       4.b-       A-       0          0          0
2       .          .        .          .          .
2.5     16a        ve       .          .          2900.5
2.75    16b-       Ma-      .          .          3406
3       4..dd      ri-      3901       .          3901
4       .          .        .          .          .
4.75    16cc       .        .          .          7344.9
=2      =2         =2       =2         =2         =2
1       4b-        a        7814.6     7814.6     7814.6
2       4r         .        .          .          .
3       2r         .        .          .          .
==      ==         ==       ==         ==         ==
*-      *-         *-       *-         *-         *-
```

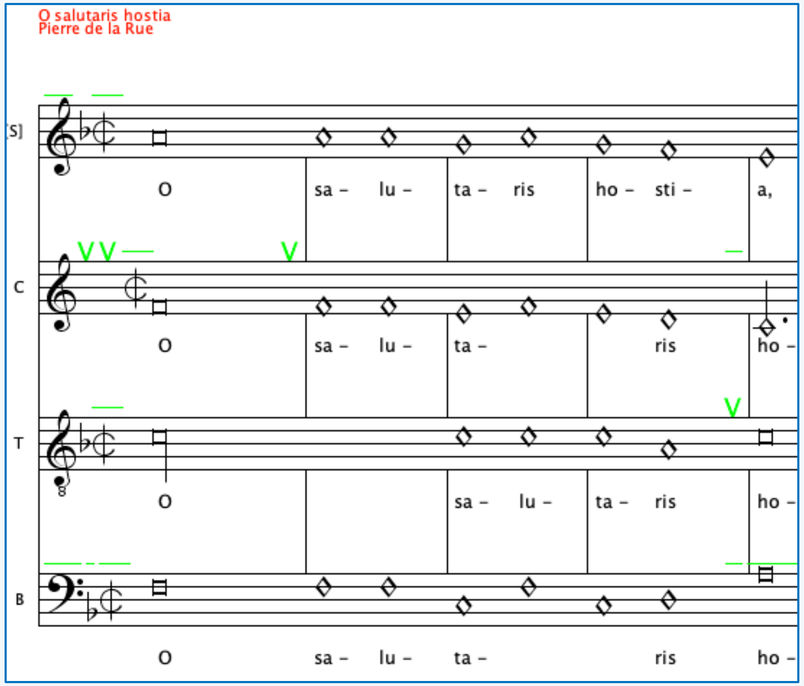
pitch made fully explicit (b-, dd)
(Devaney and Gauvin 2017)

CMME.org: transformation

- created by Ted Dumitrescu (1999)
- visual input of mensural notation, stored as XML
- transformed into CMN, in various renderings
- supports variant readings
- visual orientation of encoding

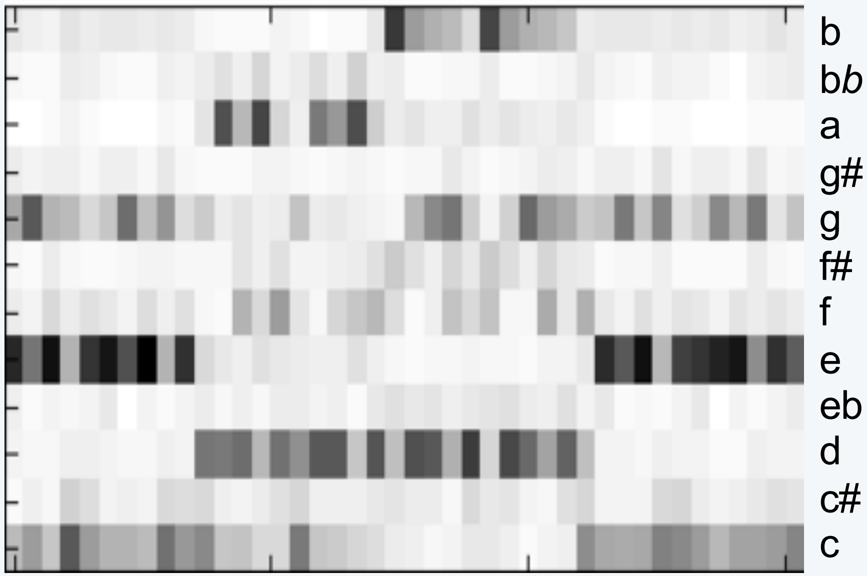
```
<Note>
  <Type>Brevis</Type>
  <Length>
     <Num>4</Num>
     <Den>1</Den>
  </Length>
  <LetterName>B</LetterName>
  <OctaveNum>2</OctaveNum>
  <ModernText>
     <Syllable>munt</Syllable>
     <WordEnd />
  </ModernText>
</Note>
```



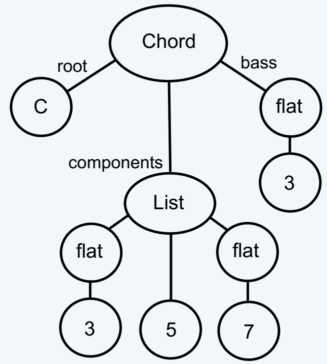


MP3 and chord labels: exploration

- MP3: perceptual compression of audio (1993)
- chord label estimation via chroma features (pitch class binning of audio)
- formal chord syntax (Harte et al. 2005)
- huge amounts of guitar tabs and chord labels on the web
- labels used in services such as Chordify
  - https://chordify.net/chords/j-s-bach-prelude-in-c-major-rousseau



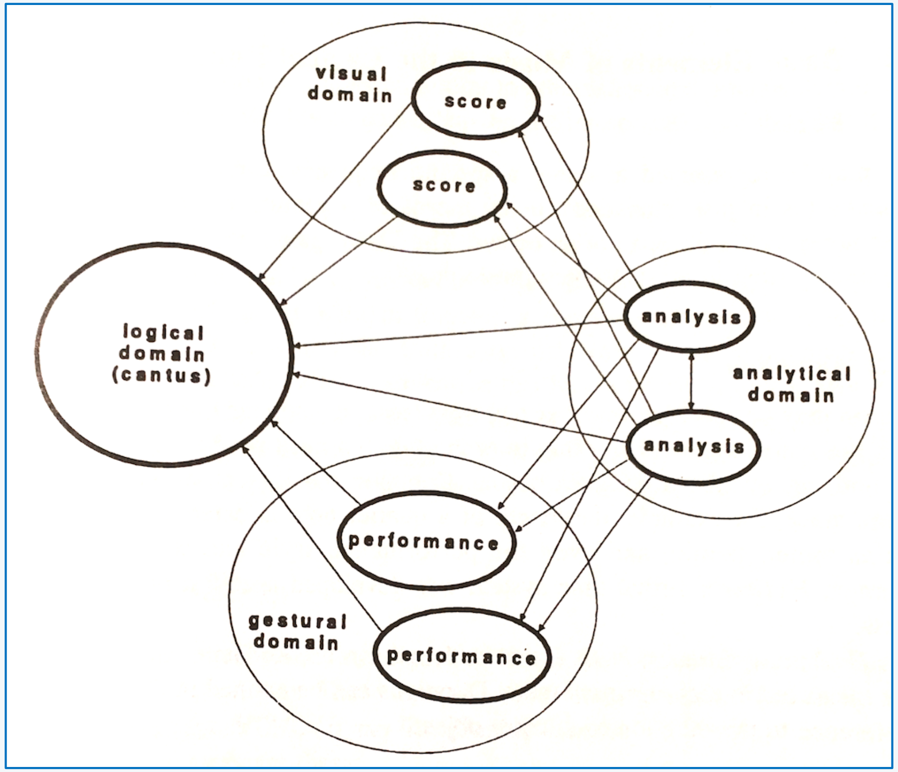
Lee & Slaney (2006): BWV 846



C:(b3,5,b7)/3

SMDL: relating

- Standard Music Description Language (Sloan 1993)
- music representation language and interchange format
- proposal for 4 interrelated domains
- *cantus* domain specifies musical logic



```
<start>
  <ce>t 1 eb
  <ce>t 2 c
  <ce>t 1 bb
  <ce>3t4 0 bb
  <ce>t4 1 g
```

```
<pitchgam id=pitchgm0 --start of pitch gamut--
  gamutdes="conventional 12-tone equal
temperament"
  highstep=11
  octratio-2 >
…
  <namestep><pitchdef><pitchnm>eb</pitchnm>
     <gamstep>0</gamstep></pitchdef></namestep>
  <namestep><pitchdef><pitchnm>f</pitchnm>
    <gamstep>2</gamstep></pitchdef></namestep>...
```

Summary

task	music as	example	other candidates
create	process	MUSIC III	Csound
preserve	heritage	PAEC	ESAC
render	product	DARMS	MusicXML, MIDI, Lilypond
analyse	problem	Humdrum	Music21
transform	source	CMME	ECOLM
explore	experience	MP3, chord labels	ABC, guitar tabs
relate	information	SMDL	IEEE 1599, MEI

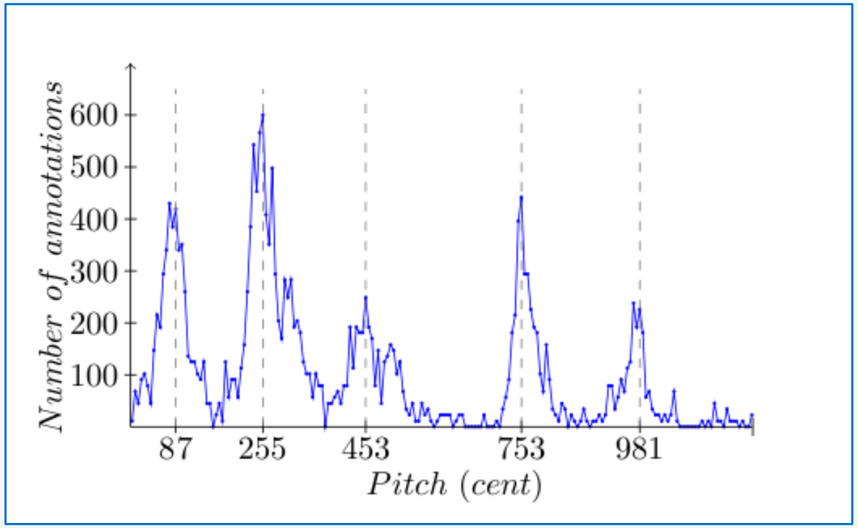
NB most systems cover a range of tasks

Pitch representations

- various viewpoints
  - freq <-> pitch
  - chromatic <-> diatonic
  - pitch <-> interval
  - graphical <-> logical
- mappings provided in e.g. Music III and SMDL

- underlying modelling issue
  - discretisation of the continuous frequency space
  - can be done in a number of ways
  - repertoire and task dependent

- in rhythm, discretisation is also an issue
  - durations unspecified in lute tablature
  - swing, inegale, phrasing…



pitch discretisation in Tarsos

Interoperability

- encoding systems are designed differently, e.g.
  - representational needs of the intended task
  - demarcation of graphical and logical
  - discrete vs. continuous
  - coverage of notation systems
  - uncertainty and incompleteness

- interoperability is not a core musicological task
- important mainly (only?) when size matters
- implies aligning academic communities
  - tend to have separate interests, goals, values and therefore encoding convictions

Where does this leave you?

suppose your work could benefit from computing, ask yourself the following questions

1. does it involve music processing?
   – if not, use general tools
2. what are the modelling requirements of the task (data, process)?
   – can you get access to suitable software?
3. if so, is data available?
   – is it suitable for the task, after preprocessing?
   – is data quality sufficient?
     - *big data:* use as is and report quality metrics
     - *small data:* correct, add missing information
     - encode essential missing items, using same format
     - donate your encodings to the community
   – if the answer is no, either *stop* or *move to next level*

Where does this leave you?

4. create your own data
   - *big data:* create proper (international) project
   - *small data:* select *lightweight* method to create *optimal* data for you
   - spend 10% of your effort to make it reusable
     - extensibility
     - documentation
     - metadata
     - quality metrics
5. distribute
   - use existing channels
   - choose appropriate license (e.g. Creative Commons)

our personal collection of music examples, transcriptions etc. is an important resource to be mobilised

Towards maturity

- many issues in computational musicology have been solved
  - representational issues are understood
  - surfeit of computational methods (often from Music Information Retrieval) ready to be explored
- data availability is the main bottleneck
  - folk song research is probably doing best
  - well-connected community

- some thoughts on data creation…
  - humanities style of collaboration
  - coordination without parochialism
  - extensible minimalism for maximal efficiency (10% rule)
  - what makes data creation fun and rewarding?
  - and yes, we should mobilise the (often very knowledgeable) citizen scientists

Thank you!

Bibliography (1)

Brinkman, A. R. (1986). Representing Musical Scores for Computer Analysis. *Journal of Music Theory*, *30*(2), 225–275.

Brook, B. S., Gould, M., LaRue, J., Bengtsson, I., Bridgman, N., Benton, R., & Chaillon-Guiomar, P. (1964). Notating music with ordinary typewriter characters (a plaine and easie code system for musicke). *Fontes Artis Musicae*, *11*(3), 142–159.

Devaney, J., & Gauvin, H. L. (2017). Encoding music performance data in Humdrum and MEI. *International Journal on Digital Libraries*, *20*, 81–91.

Dydo, S. (1991). *The Note Processor User's Guide*. Thoughtprocessors.

Erickson, R. F. (1975). "The Darms project": A status report. *Computers and the Humanities*, *9*(6), 291–298.

Galilei, V. (1584). *Il Fronimo*. Venice: Herede di Girolamo Scotto.

Harte, C., Sandler, M. B., Abdallah, S. A., & Gómez, E. (2005). Symbolic Representation of Musical Chords: A Proposed Syntax for Text Annotations. *Proceedings ISMIR*, *5*, 66–71.

Huron, D. (1988). Error categories, detection, and reduction in a musical database. *Computers and the Humanities*, *22*(4), 253–264.

Bibliography (2)

Huron, D. (1992). Design principles in computer-based music representation. In A. Marsden & A. Pople (Eds.), *Computer representations and models in music* (pp. 5–39). Academic Press Ltd.

Lee, K., & Slaney, M. (2006). Automatic Chord Recognition from Audio Using a HMM with Supervised Learning. *Proceedings ISMIR*, 133–137.

Mathews, M. V. (1961). An Acoustic Compiler for Music and Psychological Stimuli. *Bell System Technical Journal*, *40*(3), 677–694.

Selfridge-Field, E. (Ed.). (1997). *Beyond MIDI. The handbook of musical codes*. MIT Press.

Sloan, D. (1993). Aspects of Music Representation in HyTime/SMDL. *Computer Music Journal*, *17*(4), 51–59.

Valk, R. de (2015). *Structuring lute tablature and MIDI data: Machine learning models for voice separation in symbolic music representations* (PhD Thesis). City, University of London.

Wiering, F. (1997). DARMS extensions for lute tablatures. In Selfridge-Field, E. (Ed.), *Beyond MIDI. The handbook of musical codes* (pp. 201–206). MIT Press.

Wiggins, G., Miranda, E., Smaill, A., & Harris, M. (1993). A Framework for the Evaluation of Music Representation Systems. *Computer Music Journal*, *17*(3), 31–42.