Salzburg 2017

Audiveris OMR engine

herve.bitteur@audiveris.org

April 8-9, 2017

Agenda

- Project status
  - ✓ Github organization
  - ✓ Resources
- OMR data
  - ✓ Public model
  - ✓ Symbol interpretation graph (SIG)
  - ✓ Sheet pipeline
- Recognition
  - ✓ Samples repository
  - ✓ Neural networks
  - ✓ Further use of deep learning

Organization on Github

- https://github.com/Audiveris

- audiveris-eg
  - ✓ **E**arlier **G**eneration (former V4 on Kenai)
  - ✓ OMR application with basic UI
  - ✓ GPL
- audiveris
  - ✓ OMR **engine** (former private V5 on BitBucket)
  - ✓ [UI only for engine development]
  - ✓ AGPL

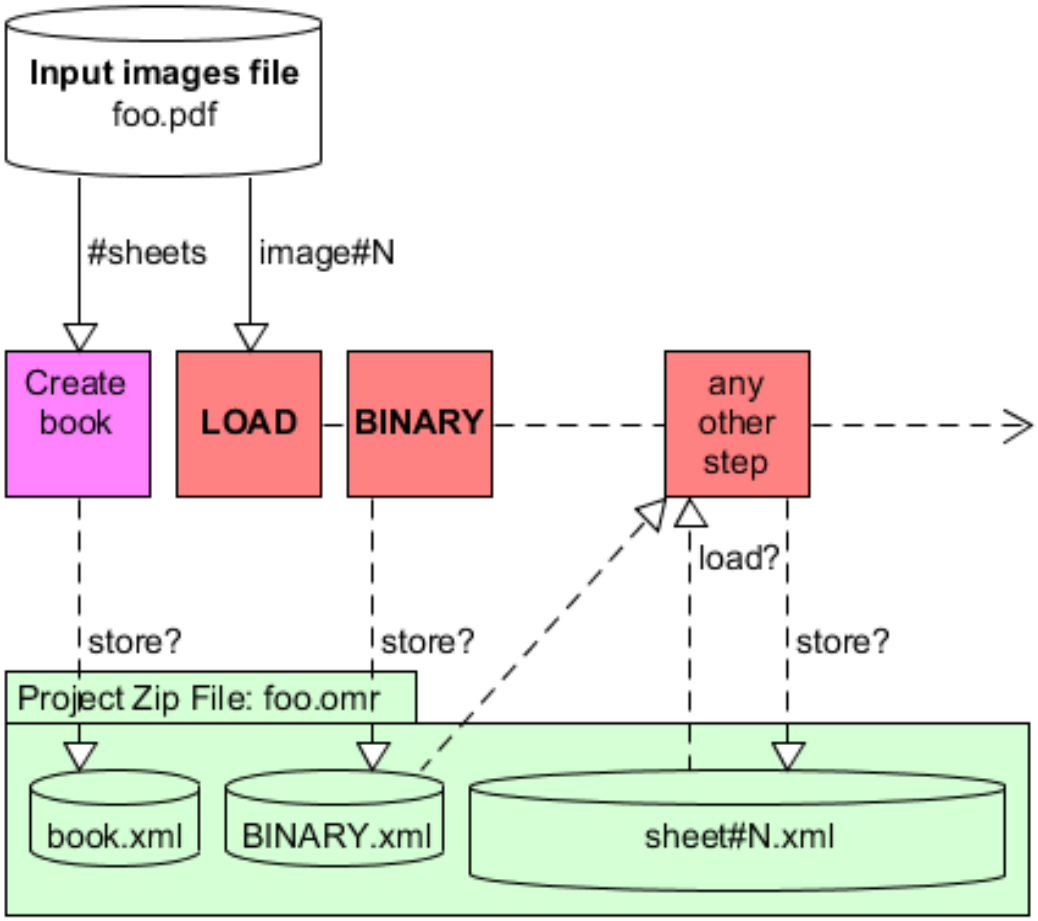
Current resources

- On board
  - ✓ Hervé Bitteur
  - ✓ Maxim Poliakovski
- Collaboration with ZHAW university
  - ✓ **Z**ürcher **H**ochschule für **A**ngewandte **W**issenschaften (= *Zurich University of Applied Sciences*)
  - ✓ « Deep Score » project
    - Philipp Ackermann
    - Lukas Tuggener
    - Thilo Stadelmann
    - Gabriel Eyyi
    - Diego Browarnik

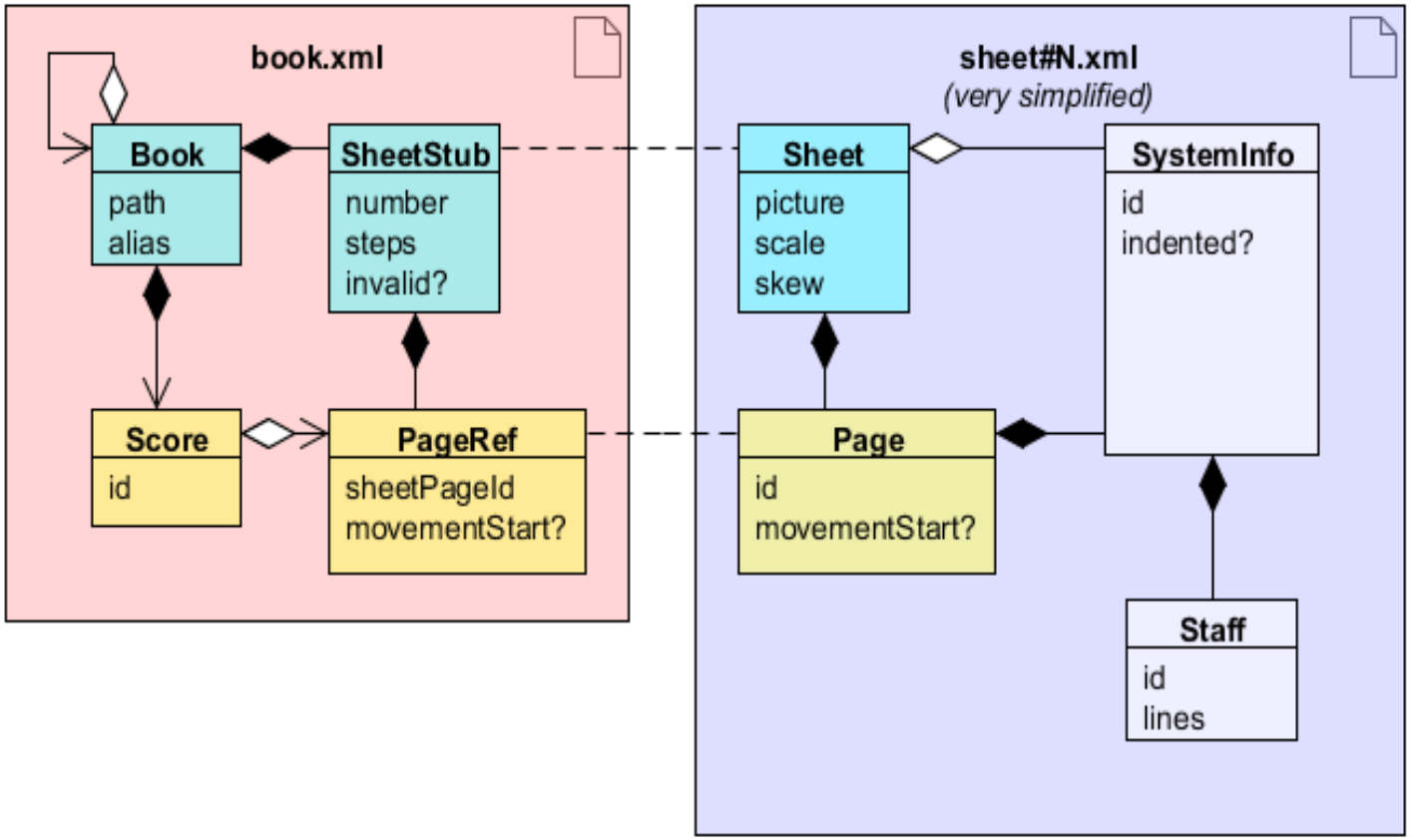
Time for a quick demo

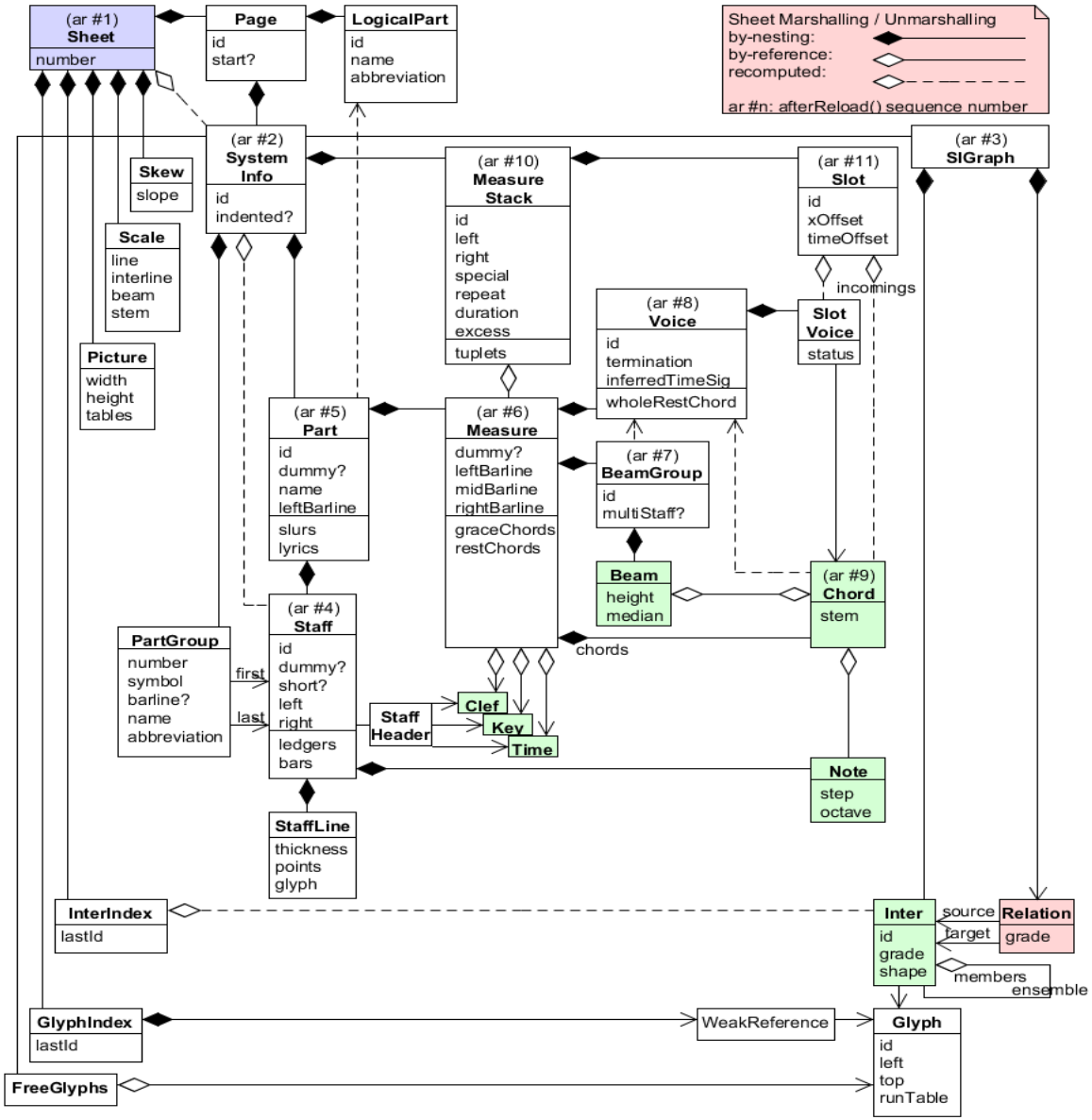
- Show sheet processing
  - ✓ Nota: Set switches for intermediate displays
- Show project file
  - ✓ Zipped archive
  - ✓ book.xml
  - ✓ sheet#n.xml

Project « .omr » file



Top containers




(ar #1)
Sheet
number
Page
id
start?
LogicalPart
id
name
abbreviation
Sheet Marshalling / Unmarshalling
by-nesting:
by-reference:
recomputed:
ar #n: afterReload() sequence number
(ar #2)
System
Info
id
indented?
(ar #3)
SIGraph
Skew
slope
Scale
line
interline
beam
stem
Picture
width
height
tables
(ar #10)
Measure
Stack
id
left
right
special
repeat
duration
excess
tuplets
(ar #11)
Slot
id
xOffset
timeOffset
incomings
(ar #8)
Voice
id
termination
inferredTimeSig
wholeRestChord
Slot
Voice
status
(ar #5)
Part
id
dummy?
name
leftBarline
slurs
lyrics
(ar #6)
Measure
dummy?
leftBarline
midBarline
rightBarline
graceChords
restChords
(ar #7)
BeamGroup
id
multiStaff?
Beam
height
median
(ar #9)
Chord
stem
chords
PartGroup
number
symbol
barline?
name
abbreviation
first
last
(ar #4)
Staff
id
dummy?
short?
left
right
ledgers
bars
Staff
Header
Clef
Key
Time
Note
step
octave
StaffLine
thickness
points
glyph
InterIndex
lastId
Inter
id
grade
shape
source
target
Relation
grade
members
ensemble
GlyphIndex
lastId
WeakReference
Glyph
id
left
top
runTable
FreeGlyphs

Sheet processing pipeline

1. LOAD : Load the sheet (gray) picture
2. BINARY : Binarize the sheet picture
3. SCALE : Compute sheet line thickness, interline, beam thickness
4. GRID : Retrieve staff lines, barlines, systems & parts
5. **HEADERS** : Retrieve Clef-Key-Time systems headers
6. STEM_SEEDS : Retrieve stem thickness & seeds for stems
7. BEAMS : Retrieve beams
8. LEDGERS : Retrieve ledgers
9. HEADS : Retrieve note heads & whole notes
10. STEMS : Build stems connected to heads & beams
11. REDUCTION : Reduce structures of heads, stems & beams
12. CUE_BEAMS : Retrieve cue beams
13. TEXTS : Call OCR on textual items
14. MEASURES : Retrieve raw measures from groups of bar lines
15. CHORDS : Gather notes heads into chords
16. CURVES : Retrieve slurs, wedges & endings
17. **SYMBOLS** : Retrieve fixed-shape symbols
18. RHYTHMS : Handle rhythms within measures
19. LINKS : Link symbols
20. PAGE : Connect systems within page

Demo

- Show samples repository
  - ✓ Organized by sheets & shapes
  - ✓ Provide sample glyph
  - ✓ [Within containing image]

Around Neural Networks

- Extraction
  - Glyph → Features
- Classification
  - Features → Shape(s)
- Ranking / Validation / Rejection
  - Based on NN output score
- Segmentation help
  - Pixels gathered into (overlapping) glyphs
  - Glyph1 → top shape 1
  - Glyph2 → top shape 2
  - Which glyph to select ?

Two NN classifiers are used

- Basic
  - ✓ Input: ART + geometric moments of glyph pixels
    - – Vector of 110 values
  - ✓ Shallow architecture (2 layers)
  - ✓ Discriminative

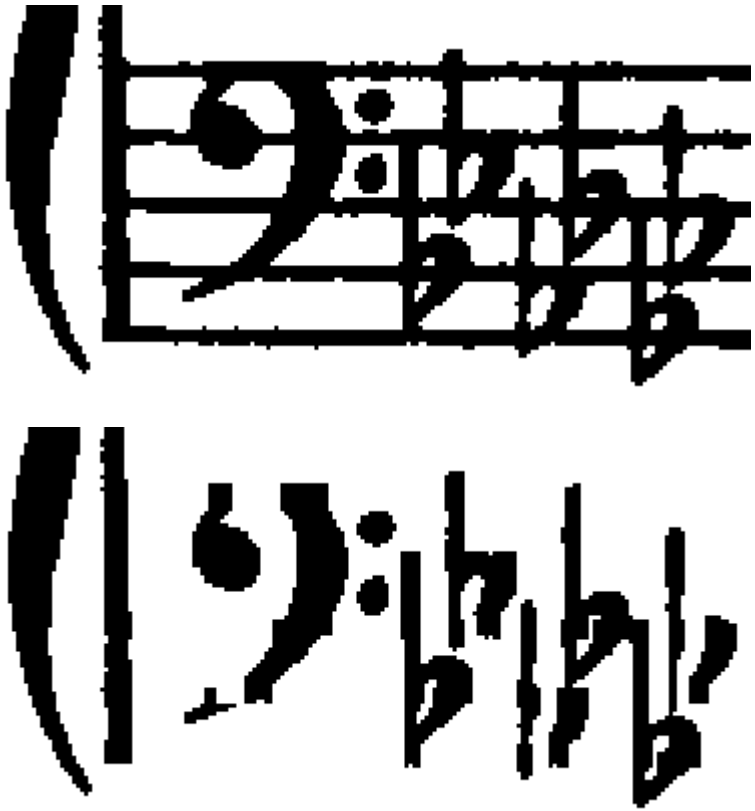
- Deep
  - ✓ Input: scaled glyph pixels
    - – Vector of 1152 values (48 x 24, interline 5)
  - ✓ Deep convolutional architecture (6 layers)
  - ✓ Very good on classification
  - ✓ Questionable on ranking (due to softmax activation?)

Segmentation problems

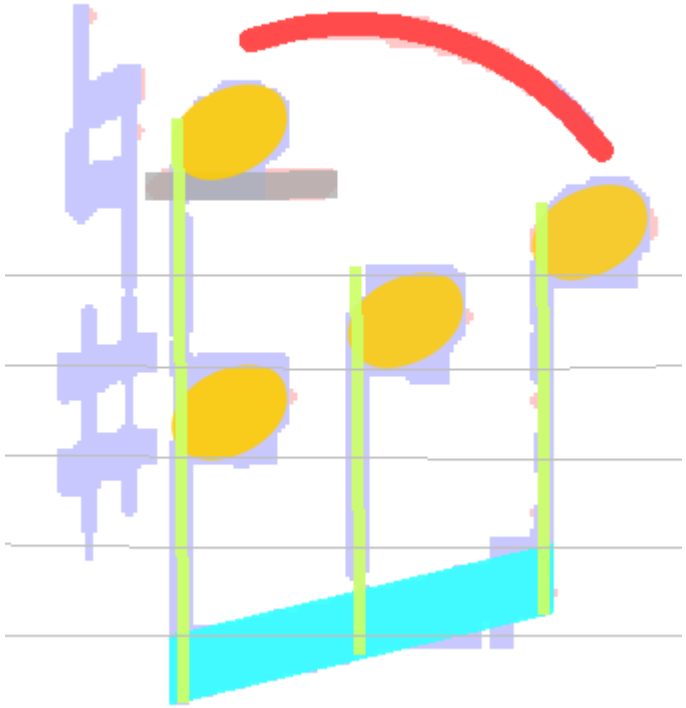
- How do we get classifier input?
- Typical sequence is:
  1) Detection & removal of staff lines
  2) Aggregation of connected pixels into glyphs
  3) Glyphs aggregation into larger compound glyphs?
  4) [Features extraction from glyph pixels]
- Symptoms
  - ✓ Collateral damage due to staff removal
  - ✓ No easy control of compound aggregations
  - ✓ No easy way to separate stuck symbols
    - – Heuristic Over-Segmentation?
  - ✓ Complex shape-specific processing

Typical examples

- Staff removal



- Stuck symbols

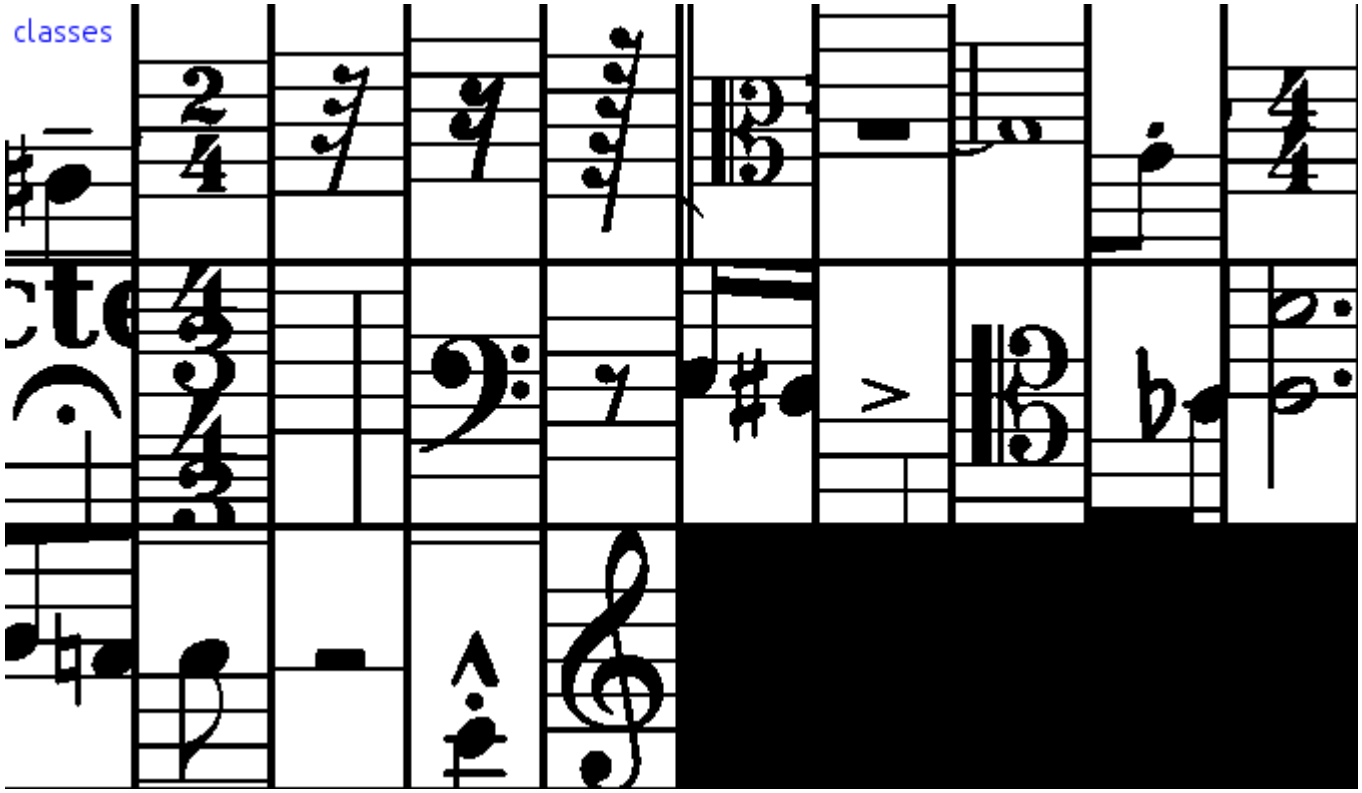


Solution to investigate?

- Train NN on context-full sub-images
  - ✓ Extract a sub-image centered on every symbol
  - ✓ Keep all pixels surrounding the symbol
  - ✓ No removal (not even staff lines)

- Expectations
  - ✓ Elegant solution to segmentation problems
  - ✓ Simplication of specific algorithms

Early ZHAW data



- Impressive recognition
  - ✓ Ratio ~99.5 %

Topic for « hack of the day »

- Use MuseScore symbolic data
  - ✓ To generate a bunch of sub-images
  - ✓ Even add some distortion?
  - ✓ The more data, the better recognition
- Scale sub-images properly
  - ✓ Using a fixed staff interline value (10 pixels?)
  - ✓ Would speed up training
  - ✓ Would provide bounding box for detected symbols
    - A way to detect overlaps
- Define and train a network on this data

Possible Audiveris integration

- Lookup valid candidate symbols
  - ✓ At every sheet location?
  - ✓ Use SIG to resolve conflicts
  - ✓ ...
- Apply first to current NN usage
  - ✓ SYMBOLS step
  - ✓ HEADERS step
- Then
  - ✓ HEADS step? (today based on template matching)

Thank you

Q & A

Audiveris [latin] := « *you will have heard* »

herve.bitteur@audiveris.org