Optimising MIR Systems Using Automatic Music Analysis Algorithms

Anna Pienimäki
University of Helsinki
Department of Computer Science
Supervisors: Prof. Esko Ukkonen, Kjell Lemström
7.10.2004

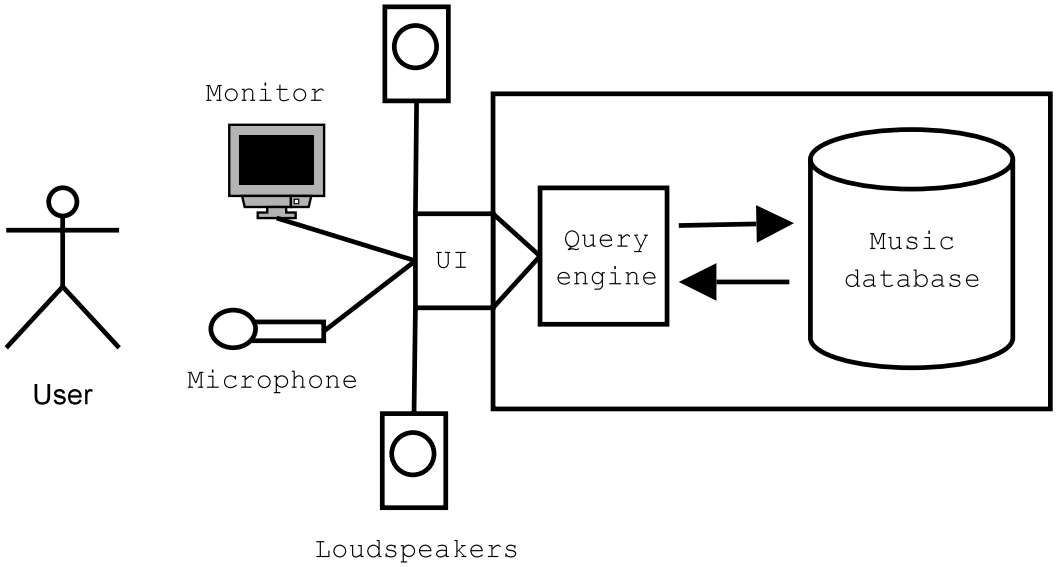
Outline

- I Personal Background
- II MIR systems (QBH systems)
- III The retrieval process
- IV What I've done this far? What I'm going to do next?

Personal Background

- PhD student in Computer Science
- MSc in Computer Science (Information systems)
- MA in Musicology (Computer-assisted musicology)
- Several bands (from indie pop to death metal)
- AE weekend warrior (recording, mixing etc.)

MIR systems (1)

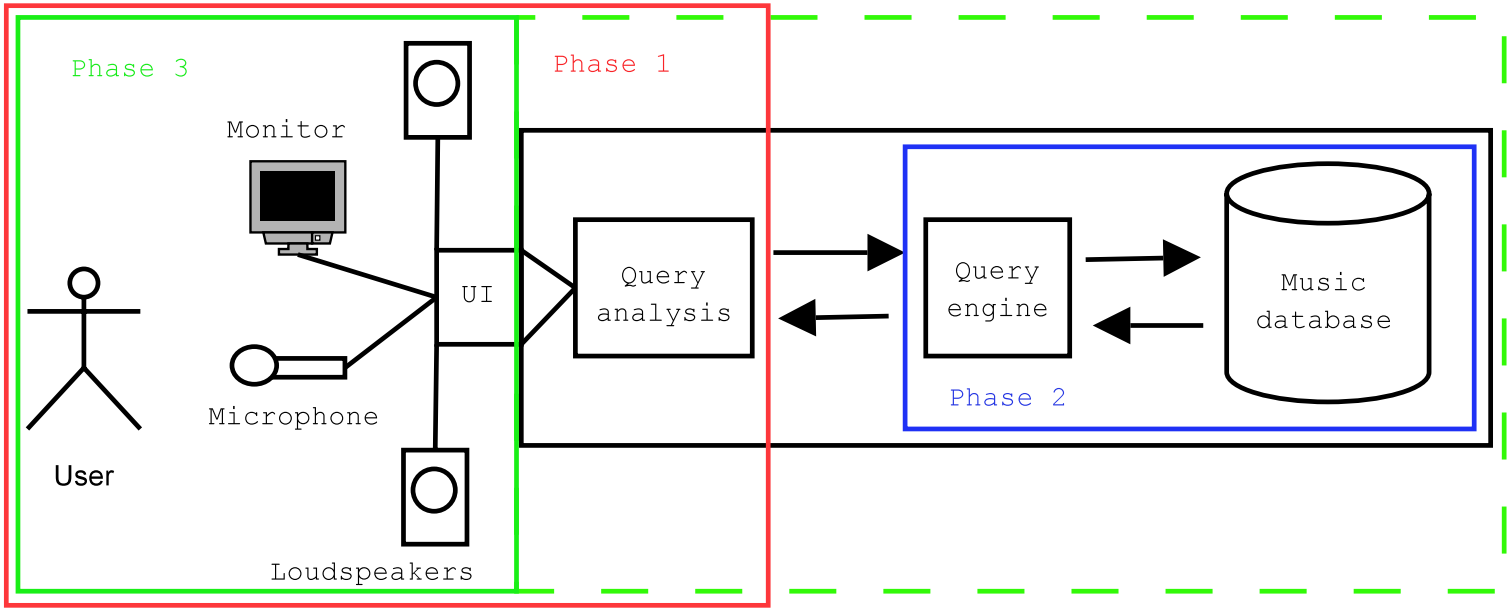


- A classic query-by-humming system

MIR systems (2)

- Several parts of the MIR systems are already at very satisfactory level:
  - Efficient search algorithms (both monophonic and polyphonic)
  - Melody transcriber (MAMI for example)
  - etc.
- What about the process?
  - I claim: The retrieval process itself could be much better!

The retrieval process (1)



- The retrieval process can be divided into at least three phases.

The retrieval process (2)

- Phase 1: Enhancing the query
  - Fact: most of the users can't sing properly
  - Solution 1: make matching algorithms more robust
  - Problem: huge amount of result candidates
  - Solution 2: do something to the query before matching
  - Idea 1: user specific profile
  - Idea 2: melodic expectancy (culture related) [PhD Thesis by T. Eerola 2003]

The retrieval process (3)

- Phase 2: Enhancing the database structure
  - Fact: database sizes are going to be huge
  - Solution 1: make simple and efficient algorithms
  - Problem: (MIDI) data isn't always perfect - there has to be a certain amount of robustness
  - Solution 2: develop efficient index structures
  - Idea 1: use repeating patterns
  - Idea 2: use clustering (extract musically meaningful melodies and cluster them)
  - Idea 3: develop a suffix-tree like structure for geometrical data

The retrieval process (4)

- Phase 3: Enhancing the browsing
  - Fact: there are several kind of needs for browsing
  - Solution 1: choose the best algorithm for every need and use that when needed
  - Problem: user's needs might change during the retrieval and browsing process
  - Solution 2: organise the database beforehand
  - Idea 1: use clustering based on several features

What I've done this far? (1)

- Indexing based on repeating patterns
  - Published in Proc. ISMIR 2002
  - Briefly: "Musical version of Ahonen-Myka algorithm"
  - Algorithm: start with 2-grams, merge them to 3-grams etc. until there aren't any n-grams left that are frequent enough. The gaps are allowed.
  - Main properties: Allows certain amount of variation. Works with polyphonic data. Is transposition invariant.

What I've done this far? (2)

- Clustering based on paradigmatic and surface level analyses
  - (will be) published in Proc. ISMIR 2004 (4 pages poster)
  - Briefly: clusters song collection based on several "musical" features
  - Algorithm: melody and results of the harmonic and paradigmatic analyses are used to build a representation of the whole song. The music collection is then clustered using these representations and a certain similarity measure.
  - Main properties: uses information on several levels.

What I'm going to do next?

- collect the real user data
- analyse it and form certain heuristics
- apply musical expectancy theory to them

Questions? Comments?

Thank you!