

Media wrangling in the car with GENIVI reqs

Collecting all your music in one place

Jonatan Pålsson

February 2, 2014



Outline

- 1 What does a media indexer actually do?
- 2 The current state of media indexing
- 3 Automotive-specific requirements

Outline

- 1 What does a media indexer actually do?
- 2 The current state of media indexing
- 3 Automotive-specific requirements

All your media in one place

Browsable, searchable, media recommendations tailored for just you!

What you really want to browse is *media metadata*

- Artist, genre, albums, etc
- Mood categorization
- Similar artists, artists grouped by geographical attributes, etc

Searching and browsing media is not cheap

- Sometimes, the metadata is hidden.. :-(
- Incorrectly used character encoding is a serious pain
- Schemes designed for extensibility are often inefficient to parse
- Some metadata needs to be computed, and not simply read
- Maybe the metadata is not even locally available?

What's commonly available locally in a media file?

Let's look at ID3

- TALB - Album
- TBPM - Beats per minute
- TCOM - Composer
- TCOM - Copyright message

What's available online then?

Open metadata databases

- MusicBrainz.org
- TheMovieDB.org
- OpenSubtitles.org

What additional information can I find about.. Regina Spektor?

- YouTube, SoundCloud, Twitter and official websites
- Place and date of birth
- Current place of residence
- A comprehensive list of releases
- .. and a bunch of other crowd-sourced information

But the information is spread all over the place! :-)

.. and this is exactly why we want media indexers! :-)

An indexer can ..

- Gather local metadata (ID3, Vorbis comments, EXIF, ..)
- Calculate metadata (mood, artist collaborations, ..)
- Gather all sorts of calculated or humanly contributed metadata
- Make all of this metadata searchable and browsable for the end user

What are the important components of an indexer, then?

- Modularity and extensibility
- Metadata querying capabilities
- Extraction efficiency
- Robustness

Outline

- 1 What does a media indexer actually do?
- 2 The current state of media indexing
- 3 Automotive-specific requirements

Let's look at what's available already..

Tracker

- The media indexer of the Nokia N9 and SailfishOS
- Integrated and usable from GNOME
- Queried using the SPARQL language
- Interfaced using D-Bus
- Written in C using Glib + GObject. There's also some Vala :)
- Highly modular extraction process
- Supports most common media formats
- Uses SQLite for storage
- Storage scheme is based on the NEPOMUK Ontologies

Let's look at what's available already..

Nepomuk-KDE

- Used in the KDE Desktop environment
- Written in C++ using Qt
- Interfaced using D-Bus
- Depends on KDE libraries :-/
- Uses the Virtuoso graph database
- Storage scheme is based on the NEPOMUK Ontologies

Let's look at what's available already..

Light media Scanner

- Developed with speed and small footprint in mind
- Written in pure C
- Simple D-Bus interface
- Used in Tizen-IVI?
- Uses the SQLite for storage and direct queries

Outline

- 1 What does a media indexer actually do?
- 2 The current state of media indexing
- 3 Automotive-specific requirements**

So.. What's this GENIVI thing?

- A consortium consisting of several large car manufacturers and OEMs
- Divided into several working groups, such as the Connectivity Expert Group or the HMI Expert group
- Aims to standardize open source components and APIs in the automotive industry
- The software stack for each new headunit should not need to be re-written from scratch!

GENIVI Requirements

- Metadata properties are categorized into Priority 1 and 2
 - P1 is essential; artist name, album, etc
 - P2 is optional; lyrics, artist date of birth, etc
- API should be D-Bus, and reusable between indexers
- Fast discovery is key!
- There is only so much disk space...
 - There is actually such a thing as too much information
 - This goes for the software as well as for the metadata
- Manual interactions must be kept to an absolute minimum

So, do these three indexers fit the requirements?

Light Media Scanner

- Some metadata properties need to be added to the indexers
- Overall, indexing is fast
- Indexes files and directories sequentially
- Small storage and software footprint
- Relies on external device management
- D-Bus API is mostly for supervision, not querying, etc

So, do these three indexers fit the requirements?

Nepomuk-KDE

- Indexing performance is not as good as Tracker or LMS
- Indexes files and directories sequentially
- Designed with desktop in mind
 - Relies on a relatively heavy database engine
 - Uses the KDE libraries extensively
- Can be fully controlled via D-Bus
- Indexes in multiple stages for improved indexing times

So, do these three indexers fit the requirements?

Tracker

- Indexing performance slightly worse than LMS
- Indexes files and directories sequentially
- Designed with desktop and embedded in mind
- Relies heavily on GLib
- Stores all data in a small-ish SQLite database
- Can be fully controlled via D-Bus
- Has its own device manager
- Gathers more data than specified by GENIVI

So, do these three indexers fit the requirements?

Conclusion

No perfect fit.. Yet.

Let the hacking begin!

Case study: Tracker

- Pelagicorean hacking project; Tracker-IVI
- Indexes too much metadata.. Luxury problem.
 - Specialized, slim extractors can be developed
- Improvements to indexing speed
 - Multi-stage extraction (a'la Nepomuk) can be introduced
 - Perceived speed can be increased by randomizing indexing order
 - Some shortcuts can be taken in the indexers, since we know exactly what we want
- Turning the small-ish database into small!
 - Optimize (or.. Well.. Thrash) the original ontologies

Conclusion: There are great possibilities here!

Looking forward

- Augmenting local metadata with online information
- Mood analysis is still not really mainstream
- APIs and specialized libraries should be generic enough to share between indexers

Resources



Tracker

<https://wiki.gnome.org/Projects/Tracker>



Light Media Scanner

<http://git.profusion.mobi/cgit.cgi/lightmediascanner.git/>



Nepomuk-KDE

<http://userbase.kde.org/Nepomuk>



NEPOMUK Ontologies

<http://www.semanticdesktop.org/ontologies/>



GENIVI

<http://www.genivi.org>



Tracker-IVI

<https://github.com/Pelagicore/tracker-ivi>

Questions?

Jonatan Pålsson

jonatan.palsson@pelagicore.com