

# Semantic collection for Amarok

## Abstract :

Nepomuk has been a great semantic framework in the recent years with many applications like Dolphin using it for managing metadata associated with files. But Amarok, still doesn't make use of the existing framework which comes bundled with any KDE distro. So, the objective of the project is to develop a Nepomuk based collection backend for Amarok which is functionally equivalent to the existing embedded MySQL backend. An earlier attempt to achieve this was made this, in GSoC 2008 but it never went on to be implemented and bundled along with the Amarok package because of reliability and performance issues. This is another attempt to make use of the excellent Nepomuk framework in Amarok and make it more semantic.

## Project :

The [Nepomuk-KDE Semantic Desktop](#) project aims to become the central storage for meta data, the one from the files itself (e.g. size, author, id3-tags) collected by Strigi, the file indexer, user entered (ratings, tags, comments) and most interesting automatically generated data provided by the KDE applications (e.g. file source, relations between files and contacts or play counts). Nepomuk provides access to all these data to all applications to help them show helpful context information and it will allow to search through the data. Nepomuk itself stores this data as RDF graph in Virtuoso.

The projects aim to develop a Nepomuk based backend collection which is equivalent to the existing embedded MySQL backend. It is a two way mechanism which allows indexed data of Nepomuk to be used by Amarok, and the metadata generated by Amarok (eg. play count, song rating) to be written back into the Nepomuk index.

Why is this project necessary?

- There would be no extra burden (scanning and indexing) on Amarok as Strigi would have already indexed the audio files and extracted meta data from the whole system. Amarok would just have to read the metadata from the index.
- This will result in a uniform rating and commenting system in KDE. Other applications can make use of this in them. Eg, If a user has rated a song in Dolphin, the same will be reciprocated in Amarok since both the applications use Nepomuk. Amarok already has a rating system in place, its just that we need to use Nepomuk to handle it.

Another example maybe during audio CD burning. For example, K3b can use the metadata generated by Amarok and populate the artist, album fields, rating etc.

- Interesting applications can be built on top of this, say using Nepoogle/NepSak a song can be searched using its lyrics ([nid3:hassynchronizedText](#)) without the help of any online web service.
- The user can tag a song with a website or to another contact, say he downloaded a song from a particular website he can tag that website or if he borrowed the song from a friend of his, he can tag that contact. This helps him remember the source of his songs. The [nid3:involvedPerson](#) and [nid3:involvedPersonContact](#) can be used for the contact and



developing the Collection backend and the Query Manager. There should be methods to write back the generated metadata into the Nepomuk index and ways to retrieve the indexed metadata from Nepomuk into Amarok.

The NepomukQueryManager can be developed into an API which can be used by plugin developers to exploit the Nepomuk backend.

- During the coding phase, I will discuss with the mentor and other members of the Amarok team on the feasibility of using taglib and strigi together for better metadata generation for better meta data generation.
- I have had experience in Nepomuk before, and have not noticed any performance issues that might crop up, but if they do, I will have contingency measures ready that will be incorporated if needed. I would need the help of the Amarok and Nepomuk teams in this case for consultation.
- The code written during the 2008 GSoC attempt will be studied if it can be reused or not. Nepomuk has come a long way since 2008 and the previous code will be tested for its relevancy in the current scenario.
- The UI need not be changed much or tinkered (as the mentor suggested) The existing UI will remain the same and complete isolation will be provided to the Nepomuk backend. The entire project will have no dependency on the UI and will not add new Qwidgets or UI elements unless absolutely required. Another advantage of this is that the user can turn off the Nepomuk backend and keep using Amarok like it was before.
- The mentor has asked for unit tests for the collection backend in the final stages of the coding phase. I do not possess expert proficiency in writing unit tests, but after a quick brush up on the basics, will come up with tests to validate the project code, esp the Collection backend.

## Outcome :

If this project is successfully completed as a plugin which can be turned on or off, then it will lead to development of new applications. They can use the indexed metadata to perform metadata based search, so queries like ' song : Stairway to heaven , date : 21<sup>st</sup> February ' can be made feasible. The user can retrieve the song he played on Christmas last year similarly.

Other use cases like CD burning etc have already been mentioned before.

## Time Line :

My college exams start in the month of May get over by the end of May and I would be able to start hardcore coding only after the completion of my exams. Since, I'm in my final year I have no external constraints and can work on the project full time for the entire duration until the pencil down phase.

1 <sup>st</sup> May to 7 <sup>th</sup> May	Go through Amarok code base. Especially the Collection abstractions
7 <sup>th</sup> May to 31 <sup>st</sup> May	Brush up my Nepomuk knowledge. University exams being the reason for the long duration.
4 <sup>th</sup> June to 9 <sup>th</sup> June	Review 2008 GsoC code for its present day relevance. Decide on

	the changes to the db schema if needed.
11 <sup>th</sup> June to 30 <sup>th</sup> June	Coding on the Nepomuk collection plugin. Decide on what goes into the database. What new metadata needs to be written into the database. Consult the Nepomuk team on the right ontologies to be used. Finish the 'write to db' phase of the project. Will have a working NepomukCollection class at the end of this phase.
2 <sup>nd</sup> July to 7 <sup>th</sup> July	Unit Tests for the NepomukCollection class
9 <sup>th</sup> July to 13 <sup>th</sup> July	Mid term evaluation
16 <sup>th</sup> July to 28 <sup>th</sup> July	Implement the NepomukQueryManager. Discuss with the mentor of a possible Amarok API to be developed in line which can be used by plugins
30 <sup>th</sup> July to 4 <sup>th</sup> August	Unit Tests for NepomukQueryManager
6 <sup>th</sup> August to 15 <sup>th</sup> August	Testing the entire project. Documentation phase. Buffer stage, consult the mentor if changes or additions have to be made.
15 <sup>th</sup> August to 18 <sup>th</sup> August	Documentation and last minute changes
20 <sup>th</sup> August to 24 <sup>th</sup> August	Pencil down phase, discussion with the mentor over the work done.

## About Me :

My name is Phalgun Guduthur, a 21 year old final year under grad majoring in Computer Science in PESIT, Bangalore India. I have experience in Python, JS, PHP, C++ and Qt. I have been an avid supporter of open source since 2years now. Have been itching to contribute to an organization with the stature of Amarok since then.

I have prior experience in open source with my college project which is hosted in KDE repos. Its a semantic resource browser on top of the Nepomuk framework. More details can be found [here](#).

I attended a conference on KDE held in Bangalore, India last year called conf.kde.in where there was a talk by Lydia Pintscher on GSoC and KDE. It helped me in applying for GSoC in 2011 but unfortunately wasn't selected. But I have learnt from my mistakes and am applying again without repeating the same mistakes.

My experience in Nepomuk and love for Amarok makes it an ideal challenge for me. Incidentally this was one of the ideas we mooted for our college project. If selected, I get to work on something I wanted to since a long time, making it an interesting and rewarding experience. I always wanted Nepomuk to go mainstream and happy to help in achieving it.

## Contact Details :

Phalgun Guduthur

irc nick : phalgun on #amarok and #nepomuk-kde

email : [phalgun.guduthur@gmail.com](mailto:phalgun.guduthur@gmail.com)

github : <http://github.com/phalgun>

time zone : UTC +530