Bruno ANGELES
Project proposal
12 November 2009

Linking jMusicMetamanager and the MusicBrainz Database – Project Proposal

Introduction

jMusicMetaManager is a tool for identifying inconsistencies in music databases. It is useful in the maintenance of both personal libraries and those of large institutions. The proposed project will add a useful feature to the software by connecting it to a database of metadata.

Project Description

A new report will be added to jMusicMetaManager, with customizable options in the application's reporting panel. The report will list matches of the music library in the MusicBrainz database, in order to suggest changes to the ID3 tags as part of the metadata cleanup operation.

MusicBrainz has an up-to-date official Application Programming Interface (API) for Python, Perl, Ruby, and C#, but the Java implementation was last updated in 2006. Libmusicbrainz-java is still available through a source control server, however. It will be used, although independent libraries such as Java MusicBrainz Library and MusicBrainz Java API are also available.

It is expected that the use of MusicBrainz metadata would be a first step in cleaning up a music library. Indeed, it would allow the user to make sure that the files' ID3 tags contain as much information as is available, before a thorough consolidation of that information is performed. An interface for applying the recommended changes will not be implemented, but would be a welcome addition.

The project's documentation will follow software engineering standards.

Milestones

The following milestones are set for the current project:

- Already done: recompile libmusicbrainz-java, make sure that its basic features are still functional, and link the library with jMusicMetaManager.
- 19 November 2009: identify the best location in jMusicMetaManager's code in which to place queries to MusicBrainz, and perform some testing of the queries
- 26 November 2009: add the basic graphical user interface (GUI) features and link them to the code performing the queries
- 3 December 2009: keep implementing the features and start looking for optimizations
- 10 December 2009: perform a cleanup of the code and start testing the features
- 17 December 2009: submit potential bugs and recommendations related to jMusicMetamanager to its author (some have been found as part of a previous presentation for this class) and finish testing the features related to the MusicBrainz link
- 21 December 2009: deliver the software, accompanying documentation, and test cases.

Bibliography

[1]McEnnis, D., C. McKay, and I. Fujinaga. 2006. Overview of OMEN. In *Proceedings of the International Conference on Music Information Retrieval*. Victoria, Canada. 7–12.
http://ismir2006.ismir.net/PAPERS/ISMIR06145_Paper.pdf

The authors describe OMEN's Master-Library-Worker nodes architecture, as well as legal issues related with the sharing of music and metadata. jMusicMetaManager is presented as a possible tool to clean up information before it is sent from the Library Nodes to the Master Node.

[2]McKay, C., D. McEnnis, and I. Fujinaga. 2006. A large publicly accessible prototype audio database for music research. In *Proceedings of the International Conference on Music Information Retrieval*. Victoria, Canada. 160–3.
http://ismir2006.ismir.net/PAPERS/ISMIR067_Paper.pdf

This paper presents Codaich, a music database designed for music research. It is presented as a database that overcomes copyright restrictions and offers varied content for tools such as jMusicMetaManager.

Links

MusicBrainz. 2005. Libmusicbrainz – MusicBrainz.
http://musicbrainz.org/doc/libmusicbrainz (accessed 12 November 2009)

MusicBrainz. 2005. Source control access to libmusicbrainz-java.
http://svn.musicbrainz.org/libmusicbrainz-java/trunk (accessed 12 November 2009)