Combining multi-level audio descriptors via web identification and aggregation

Jun Wang
National Network New Media Engineering Research Center, Institute of Acoustics, Chinese Academy of Sciences
wangjun@dsp.ac.cn

Xavier Amatriain
Telefonica Research
xar@tid.es

David Garcia Garzon
Barcelona Media Centre d'Innovació, Universitat Pompeu Fabra
dgarcia@iua.upf.edu

Jinlin Wang
National Network New Media Engineering Research Center, Institute of Acoustics, Chinese Academy of Sciences
wangjl@dsp.ac.cn

ABSTRACT
In this paper, we present the *CLAM Aggregator*, a tool included in the CLAM framework that allows combining multi-level audio descriptors. The tool includes a reliable method to identify the user's local music collection using open data resources and allows users to configure, aggregate and edit music information ranging from low-level descriptors to any metadata from the semantic Web. All these steps are designed in a flexible, graphical, and user-adaptive way.

Categories and Subject Descriptors
H.4 [Information Systems Applications]: Miscellaneous

General Terms
Music Descriptors

Keywords
Annotating Tool, Linked Data, Music Descriptors, Semantic Web, XML

1.INTRODUCTION
Music feature extraction is one of the most crucial procedures in fields of Music Information Retrieval (MIR). The traditional solution is to automatically extract relevant features from the original music signal. Technologies such as Web 2.0 are making the World Wide Web a primary host of a sizeable amount of text-based and semantic information. This is allowing researchers to start to bring up novel ways to use this data for information retrieval tasks. For instance, research teams of Levy [1], Knees [2] and Barrington [3] etc. investigated metadata from the Web, including social tags, pages via Google queries and web-mined documents as an additional source of semantic metadata for MIR. All these works report that the combination of signal-level and web-based features outperforms each of them in isolation.

Nevertheless, web-based information is hardly ever associated with the audio files it refers to. The *CLAM Aggregator* provides a convenient tool for combining multi-level music information. It interlinks all levels of descriptors ranging from the audio-based frame level features, to the web-based song level descriptors. It enables researchers to deal with multivariate data analysis and develop methods to leverage the semantic gap.

The most important features of the *CLAM Aggregator* are: (i) It is being developed in the context of the *CLAM* framework [4] and *CLAM Annotator* [5], which allows to extend it in unlimited ways by embedding user-defined automatic extraction algorithms in it; (ii) It offers reliable identifications to link the audio files with the open-link Web, avoiding the obstacles of noisy, redundant or false extraction in the existing technologies; (iii) It uses standard XML language to represent data. Thus, it works with readable and understandable data files that are easily connected to external applications or databases and can also be easily converted to semantically rich RDF resources and linked back to the open-link Web.

2.FEATURES AND COMPONENTS

2.1Annotator Schemas and Descriptors
The *CLAM Annotator* (see [5]) offers a convenient GUI that allows editing multi-level descriptors. It makes use of different XML files: the *Annotator Schema* files define the list, the type and the value range of different descriptors; the *Descriptors pool* files contain all the actual values for the descriptors.

We can define multi-level descriptors in the Schema. A song-level descriptor is unique within the song scope. It can be of any type. On the other hand, a low-level descriptor has a segment or frame scope and must be of floating point type.

Descriptors may be generated by any third-party extractors by providing a proper Schema. Any extraction algorithm may dump its results in the XML representation format of a *CLAM Descriptors Pool*, without having to worry about formatting issues.

2.2Aggregation
More and better algorithms for automatic feature extraction are constantly being developed in the MIR field. The combination of different algorithms and features allows for the development of new applications. It is therefore important to allow the combination of descriptors in a single Annotator project.

The *CLAM Aggregator* offers a dynamic GUI to achieve this. The tool can itself be considered as another extractor, which also makes use of the fore-mentioned *Schema* files and *Descriptors pool* files. It extracts selected descriptors from the output pool files of different extractors, according to the user configuration in the GUI. An example of the generated configuration file is shown in listing 1. The list of *sources* defines the IDs of different

extractors (e.g., “ClamCore”, “Chord”, “SemWeb”), configures the *Schema* files, the *Descriptors pool* file suffix, and the extractors respectively. The list of *map* defines the selected attributes of each extractor, and maps the scope::attribute names from the sources to the target aggregated pool files. According to these configurations, the Aggregator’s *Schema* file and *Descriptors pool* file are automatically generated.

```
sources = [     ("ClamCore", FileMetadataSource(path=".", schemaFile="CLAM.sc",
                                poolSuffix=".example", extractor="ClamExtractor")),
                ("Chord", FileMetadataSource(path=".", schemaFile="Chords.sc",
                                poolSuffix=".chord", extractor="ChordExtractor")),
                ("SemWeb", FileMetadataSource(path=".", schemaFile="SemWeb.sc",
                                poolSuffix=".webpool", extractor="SemWebExtractor")),
]
map = [         # ('TargetScope::TargetAttribute', 'ID', 'SourceScope::SourceAttribute'),
                ("Song::Danceable", " ClamCore ", "Song::Danceability"),
                ("Song::ChordFrames", "Chord", "Song::Frames"),
                ("ChordFrame::ChordHartePcp", "Chord", "Frame::HartePcp"),
                ("Song::TagJamendo", "SemWeb", "Song::TagJamendo"),
                ("Song::Review", " SemWeb ", "Song::Review"),
                ("Song::Rating", " SemWeb ", "Song::AlbumRating"),
]
```

Listing 1. Configuration File generated from the CLAM Aggregator GUI

2.3Web Identification

Web-based descriptors are becoming more and more important in MIR fields nowadays. However, existing extraction solutions like Google query and other search-engine-based methods cannot overcome the issues related to noisy, redundant or false extractions.

The *CLAM Agreggator* includes a reliable and extendable web-based extractor. The extractor makes use of GNAT (Yves, et al., [6]) to find corresponding related identifiers of local audio files. GNAT uses audio fingerprinting and available metadata to identify the songs MBID on MusicBrainz, and then outputs RDF statements representing the links between local songs and the remote web identifiers. The proposed graph matching algorithm allows GNAT to be robust even with inaccurate or incomplete local metadata.

Using the MBID and the RDF statements, the extractor crawls through several Linking Open datasets and extracts high-level descriptors such as editorial metadata, user comments, genre, album reviews, or tags. This approach is related to the interlinking of music-related datasets on the Semantic Web [7]. The extraction could be extended in flexible ways, to respond to the need of different tasks and applications.

3.APPLICATION GUI

3.1Configuration

Extractor names, corresponding scopes and attributes can be shown graphically (see fig. 1). Users can select desired attributes by checking them. A configuration file is then generated. Furthermore, and according to the configuration, an aggregated description Schema is automatically constructed (fig. 2).

3.2Editing High-level Descriptors

Fig. 3 depicts the GUI for browsing and editing high-level descriptors. Users can enter free text, e.g. human-knowledge descriptors like “extreme sports” to describe the *Usage* attribute of a song.

3.3Fine Tuning Low-level Descriptors

Part of the powerful GUI of *CLAM Annotator* for fine tuning low-level descriptors is shown in fig. 4. Other applications for tasks such as editing segmentation marks or auralizing annotations are also provided in this framework (see details in [5])

4.CONCLUSIONS AND FUTURE WORK

In the paper we have presented the *CLAM Aggregator*, which can be used to link local music collections with the open Web, and combine multi-level descriptors in a user-configurable way. This tool is available as open source at http://clam-project.org/clam/trunk. The demo and screenshots are on Planet Clam at http://clam-project.org/planet/index.html. More screenshots can be viewed on Wikis at http://clam-project.org/wiki/Development_screenshots.

Future plans include research into the advantages such multi-level combination brings for automatic high-level semantic annotations (e.g. emotions) of music. Secondly, a conversion of the XML-based local audio descriptor files to Semantic Web resources, and linking them to corresponding identifiers on the Linking Open Data Web.

5.ACKNOWLEDGMENTS

Our thanks to CLAM development team for their contributions and the research team of Yves Raimond, etc., for allowing us to adopt tools they had developed. Part of this work has been supported by the Google Summer of Code 2008 project.

6.REFERENCES

[1] M. Levy, M. Sandler. A semantic space for music derived from social tags, In Proceedings of ISMIR 2007.

[2] P. Knees, T. Pohle, M. Schedl and G. Widmer. A Music Search Engine Built upon Audio-based and Web-based Similarity Measures. In Proceedings of SIGIR 2007.

[3] L. Barrington, M. Yazdani, D. Turnbull, G. Lanckriet. Combining feature kernels for semantic music retrieval, In Proceedings of ISMIR 2008.

[4] X. Amatriain. CLAM: A framework for audio and music application development. IEEE Software, vol. 24, no. 1, pp. 82-85, Jan./Feb. 2007, doi:10.1109/MS.2007.8.

[5] X. Amatriain, J. Massaguer, D. Garcia, I. Mosquera. The CLAM Annotator: a cross-platform audio descriptors editing tool. In Proceedings of ISMIR 2005.

[6] Y. Raimond, C. Sutton, and M. Sandler. Automatic interlinking of music datasets on the Semantic Web. In Proceedings of IDOW’08 (Beijing, China, April 2008).

[7] C. Bizer, T. Heath, D. Ayers, Y. Raimond. Interlinking open data on the web. In Demonstrations Track, 4th European Semantic Web Conference, Innsbruck, Austria, 2007.

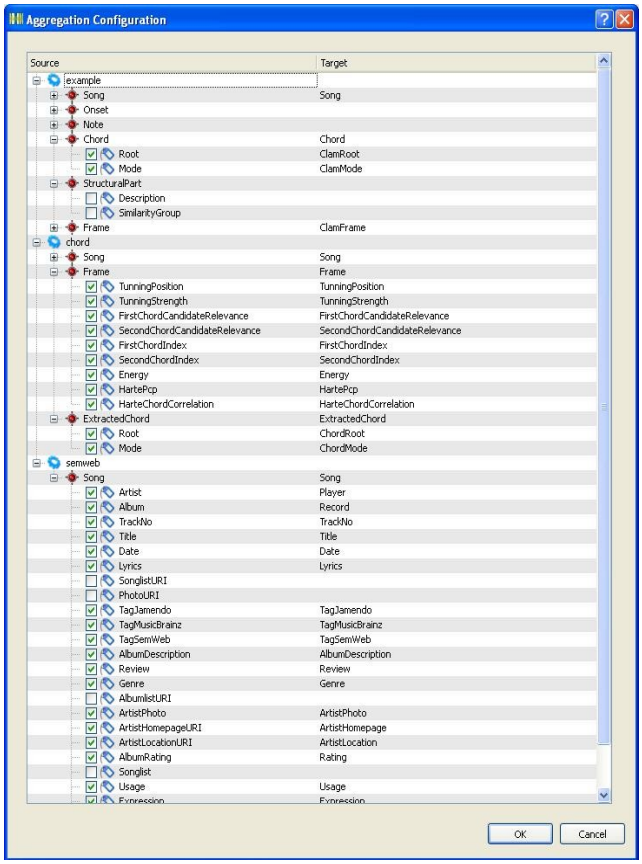

Figure 1. Configuration GUI for aggregation.

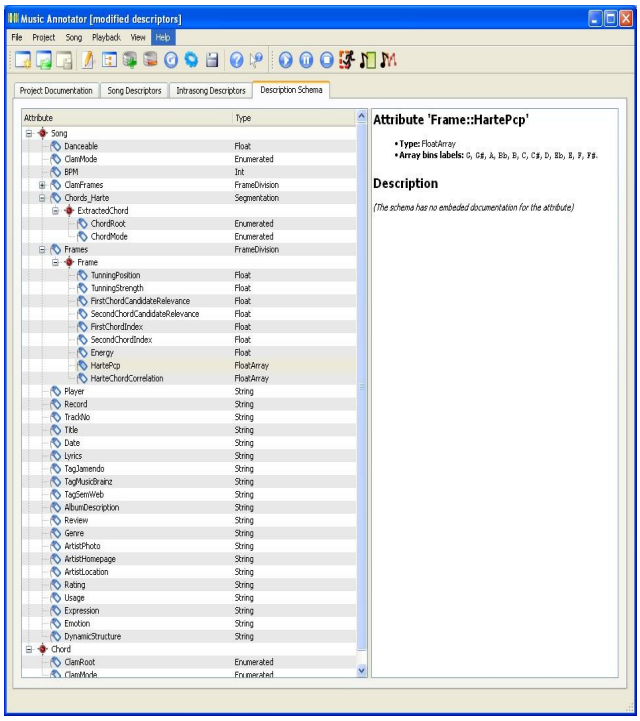

Figure 2. Aggregated description Schema.

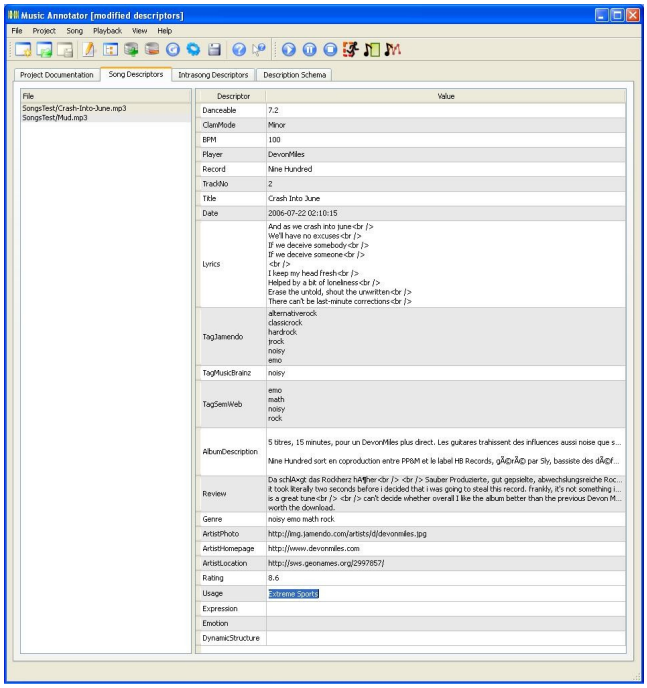

Figure 3. Editing high-level descriptors.

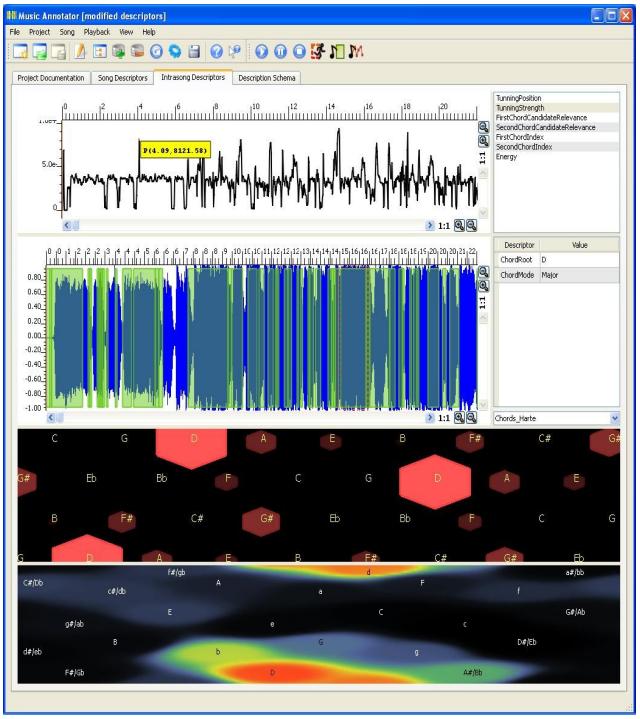

Figure 4. Fine tuning low-level descriptors.