Enabling Access To Sound Archives Through Integration, Enrichment and Retrieval: The EASAIER Project

Christian Landone, Josh Reiss, Joe Harrop*
*Centre for Digital Music, Queen Mary University of London, UK. *Royal Scottish Academy of Music and Drama, Glasgow, UK.*

The EASAIER Project

Many digital sound archives suffer from problems concerning on-line access: sound materials are often held separately from other related media, they are not easily browsed and little opportunity to search the actual audio content is provided.
The EASAIER project aims to alleviate these problems, offering a number of solutions to support sound archive managers and users. EASAIER will enable enhanced access to sound archives, providing multiple methods of retrieval, content enrichment as well as advanced visualisation and processing tools.
The system is designed according to the needs and wishes of leading European libraries, museums, broadcast archives and music schools. The Consortium has been working with these institutes to develop tools in direct response to user demands.

The EASAIER Consortium consists of 3 companies and 4 academic institutions:

Silogic - France | **QMUL – United Kingdom** | **DIT - Ireland** | **DERI - Austria**

RSAMD – United Kingdom | **NICE Systems - Israel** | **Applied Logic - Hungary**

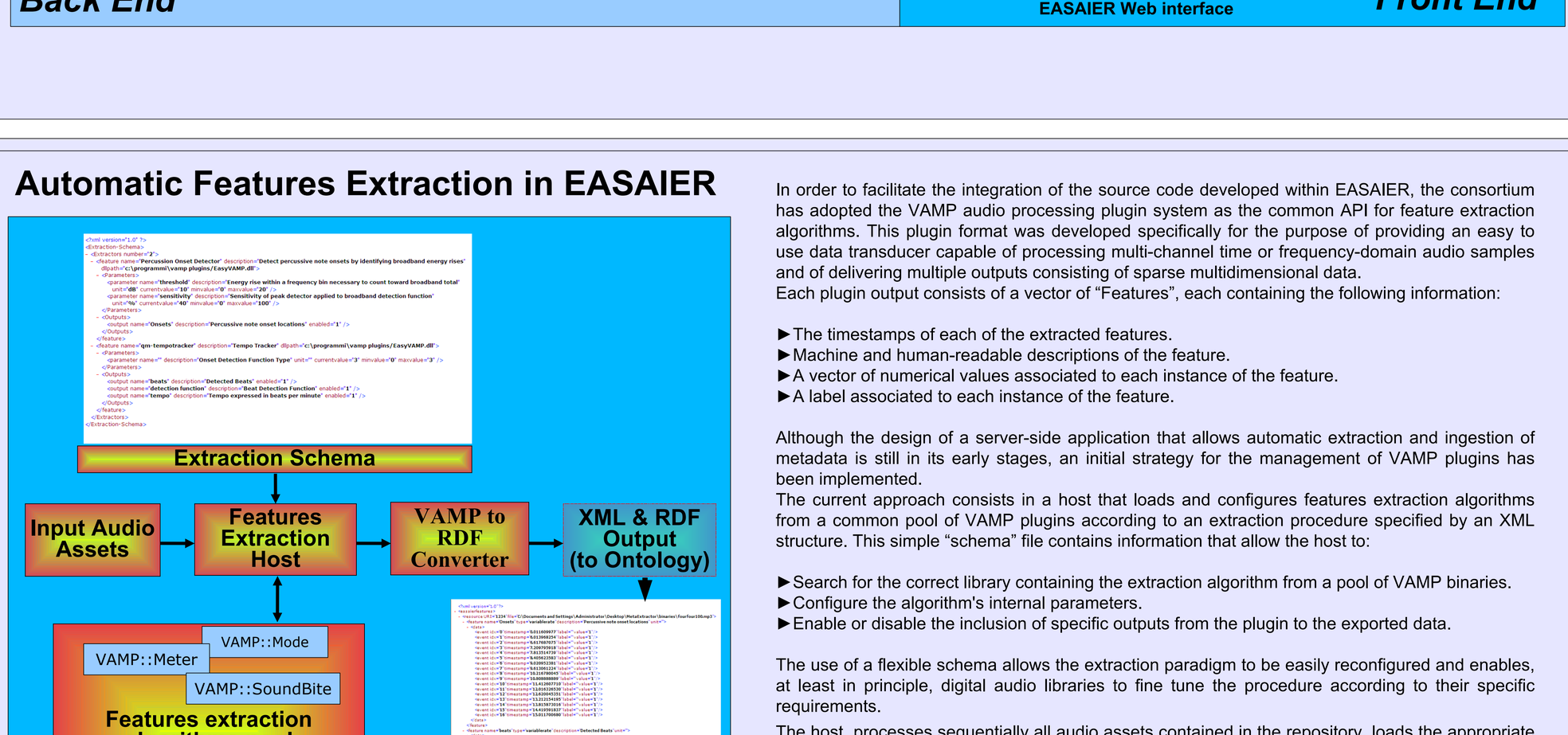
EASAIER is Funded by the European Union, Information Science Technologies EU-FP6-IST-033902

System Functionality

►**Search for media fitting a wide variety of metadata:** The system will allow searches based on various types of descriptors, including, where possible, semantically meaningful features that are automatically extracted from the audio content, and will provide the means to search across multiple and sometimes ambiguous selections of metadata.
►**Relevant near neighbour query-by-example searches:** The system can provide a ranked list of related audio assets which may be similar in musical qualities, such as timbre and structure.
►**Enriched access and interaction tools:** A set of tools that simplify the task of marking points of interests on the audio waveform is included in the front end. Instruments that allow real-time manipulation of audio during playback, such as time scaling and sound source separation can greatly aid the analysis process. In addition to signal processing capabilities, advanced visualisation tools allow the user to display a number of different pieces of data overlaid on one another and aligned according to time.
►**Choice of interfaces**: The system provides two means of accessing a repository:
●A simple web interface that can be used to perform the most basic search and retrieval operations and provides limited playback functionality.
●An advanced cross-platform application developed using the QT4 toolkit that allows comprehensive browsing, visualisation and audio processing.

EASAIER System Architecture

The EASAIER system exhibits a standard client/server architecture and will consist of five main components:
►***Binary Store***: where all assets are stored; in most cases, EASAIER will interface with established media repositories.
►***Archive Administration Application***: Allows content managers to enter assets in the archive and to process assets that were already stored prior to EASAIER's deployment. The tool will be generally automated,requesting intervention only in the instance of unreliable or ambiguous cases. The generated metadata will be saved in a separate repository as RDF triples.
►***RDF Store:*** RDF provides a framework for the description of resources as well as their metadata. The goal of RDF is to define a grammar and structure such that machines can process the knowledge being represented. As this structure is very generic, an ontology is used to define a common vocabulary.
►***SPARQL Endpoint***: SPARQL is a query language adapted to the semantic structure of RDF. It provides a suitable query interface in order to access the knowledge held by the system's internal ontology.
►***EASAIER Clients***: The applications will allow end users to access the media repository, perform queries and retrieve assets. There will be two versions of the client: a web *client, usable for off-site access* and an *advanced client* that will provide a full set of access tools



Automatic Features Extraction in EASAIER

Automatically extracted features from audio assets play a major role within the EASAIER project. The system is designed to use algorithms that extract two types of features:

►*Information extracted at fixed sample times along the asset's timeline.*
These features are commonly referred to as “low-level descriptors” and represents data that is “machine-readable”, used for the generation of abstract models that enable the system to perform searches based on metrics such as timbral and harmonic similarity, voice-print similarity and phoneme type

►*Information extracted at non-uniform sample times from the asset's timeline.*
These features are usually referred to as “Mid/High level descriptors” and consist of “human readable” data that describe music and speech events (e.g. a beat, an instrument solo, a tempo change, the gender of a speaker)
These features can be directly used to perform parametrized searches based on specific musical and speech descriptors; typical query examples include but are not restricted to:
●finding audio assets that contain exclusively musical audio or speech
●finding musical segment that exhibit a specific tempo, meter, key, orchestration or contain a particular class of instruments.

In order to facilitate the integration of the source code developed within EASAIER, the consortium has adopted the VAMP audio processing plugin system as the common API for feature extraction algorithms. This plugin format was developed specifically for the purpose of providing an easy to use data transducer capable of processing multi-channel time or frequency-domain audio samples and of delivering multiple outputs consisting of sparse multidimensional data.
Each plugin output consists of a vector of “Features”, each containing the following information:

►The timestamps of each of the extracted features.
►Machine and human-readable descriptions of the feature.
►A vector of numerical values associated to each instance of the feature.
►A label associated to each instance of the feature.

Although the design of a server-side application that allows automatic extraction and ingestion of metadata is still in its early stages, an initial strategy for the management of VAMP plugins has been implemented.
The current approach consists in a host that loads and configures features extraction algorithms from a common pool of VAMP plugins according to an extraction procedure specified by an XML structure. This simple “schema” file contains information that allow the host to:

►Search for the correct library containing the extraction algorithm from a pool of VAMP binaries.
►Configure the algorithm's internal parameters.
►Enable or disable the inclusion of specific outputs from the plugin to the exported data.

The use of a flexible schema allows the extraction paradigm to be easily reconfigured and enables, at least in principle, digital audio libraries to fine tune the procedure according to their specific requirements.

The host processes sequentially all audio assets contained in the repository, loads the appropriate features extraction algorithms and outputs data to a module that converts the internal VAMP format into an XML structure as well as the RDF description used by the EASAIER ontology.
The RDF description of events detected on each asset contain information concerning the asset's URI, a measure of the reliability of the extraction process, the type of detected event, its position and extent on the timeline and, if applicable, its associated numerical values and labels.