THE MUSIC INFORMATION RETRIEVAL EVALUATION EXCHANGE "DO-IT-YOURSELF" WEB SERVICE

Andreas F. Ehmann **J. Stephen Downie** **M. Cameron Jones**
International Music Information Retrieval Systems Evaluation Laboratory
The Graduate School of Library and Information Science
University of Illinois at Urbana-Champaign
{aehmann, jdownie, mjones2}@uiuc.edu

ABSTRACT

The Do-It-Yourself (DIY) web service of the Music Information Retrieval Evaluation eXchange (MIREX) represents a means by which researchers can remotely submit, execute, and evaluate their Music Information Retrieval (MIR) algorithms against standardized datasets that are not otherwise freely distributable. Since its inception in 2005 at the International Music Information Retrieval Systems Evaluation Laboratory (IMIRSEL), MIREX has, to date, required heavy interaction by IMIRSEL team members in the execution, debugging, and validation of submitted code. The goal of the MIREX DIY web service is to put such responsibilities squarely into the hands of submitters, and also enable the evaluations of algorithms year-round, as opposed to annual exchanges.

1. INTRODUCTION

MIREX [1] represents a community-based effort for the standardization of datasets and metrics used in the evaluation of MIR algorithms. Due to the litigious nature of copyright protected materials, MIREX has adopted a paradigm whereby community-developed algorithms are submitted to a single, central location (IMIRSEL) where they are evaluated against common, non-distributable datasets. In previous iterations of MIREX, the execution and evaluation of submitted code was handled by IMIRSEL team members. The goal of the MIREX DIY web service is to allow researchers to submit, debug, execute, and evaluate their algorithms remotely via a web interface.

The MIREX DIY service is largely built upon the Data-to-Knowledge web service (D2KWS) [3] and Music-to-Knowledge (M2K) [2] libraries. As with the majority of past MIREX evaluations, the necessary file input-output, algorithm execution, and algorithm evaluations are carried out within D2K/M2K programs referred to as itineraries. A typical itinerary used in MIREX evaluations can be seen in Figure 1.

Each component of an itinerary is referred to as a module. In general, the itineraries used for the execution and evaluation of algorithms are comprised of four types of modules. Input modules specify the locations of a dataset's audio or MIDI files, and their corresponding ground truths. File reader modules are task specific modules for reading a task's specified file format. External Integration modules are responsible for executing external code (namely the submitted algorithm). Finally, evaluation modules are task specific modules that measure the performance of an algorithm's output compared to the pre-established ground truth.

The MIREX DIY web interface is largely responsible for generating and populating the itineraries and their modules with pertinent information, thus absolving the user of this burden. The D2KWS queues and distributes jobs to sandboxed and firewalled D2K servers for execution. Real-time status and debugging information is displayed to the user, and upon successful completion, evaluation results are made available.

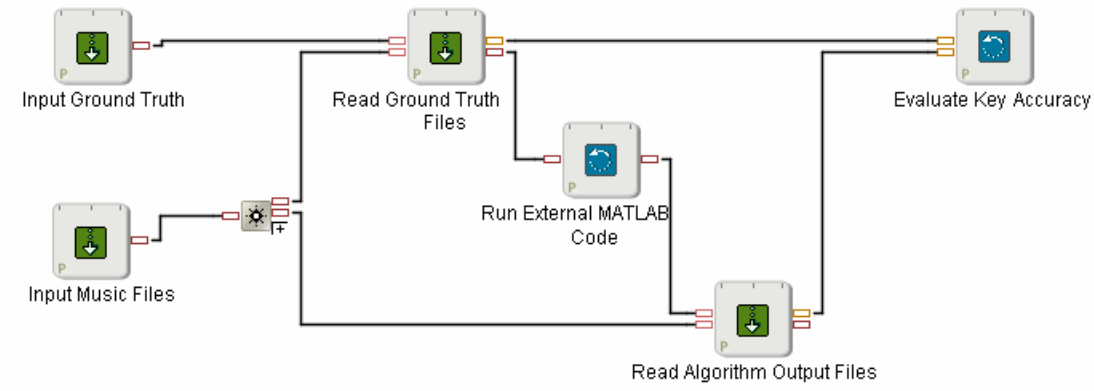

Figure 1. An itinerary used in MIREX evaluations, in this case, music key finding.

© 2007 Austrian Computer Society (OCG).

2. MIREX DIY FRAMEWORK[1]

An overview of the general MIREX DIY framework can be seen in Figure 2.

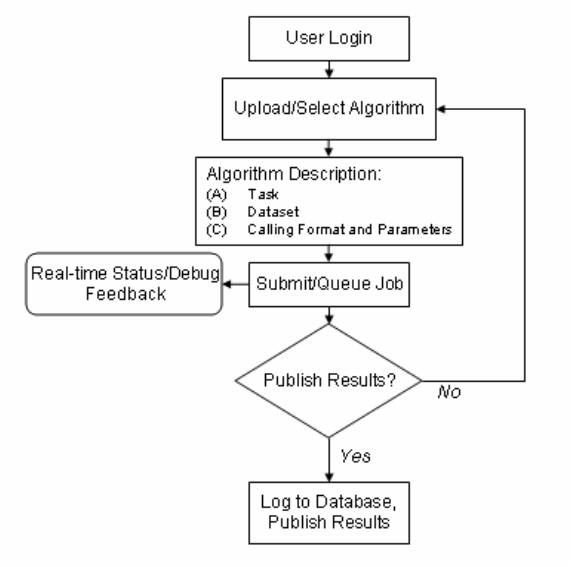

Figure 2. MIREX DIY framework overview.

Upon logging in, a user can either upload a new algorithm, or select one of his or her previously uploaded algorithms for testing. Next, the user specifies essential information for the execution of the algorithm:

(A) The user must specify a task that the algorithm is designed for (e.g., audio onset detection, audio key finding, etc.). This information is used to generate an itinerary with the task-specific file reader and evaluation modules.

(B) The user selects from the available datasets appropriate for the task selected in (A). For each task, there also exists a small test/validation dataset so the user can verify that a new upload functions correctly. The dataset selection information is used to populate the evaluation itinerary's input modules with the local paths to a dataset's audio or MIDI files and their ground truths.

(C) Information detailing the calling format and parameters is specified by the user, which subsequently populates the necessary fields in the M2K external code integration module.

After specifying the above information, the algorithm can be run. The job is queued by the D2KWS, and real-time information displaying the progress is shown to the user. Upon successful completion, the evaluation results are made available, and appropriate entries are made into a database. The user has the option to publish the results of the run. If the results are to be published, a web page containing the results of all published runs is updated.

[1] See http://music-ir.org/mirexdiy to access a limited demo version of the framework.

3. CHALLENGES

3.1. Datasets

The ability to evaluate algorithms year round raises some concern over the possibility of over-fitting to specific datasets. It may be beneficial to maintain separate datasets that are only made available for once-per-year evaluations. However, facilities for constructing unique datasets 'on the fly' from IMIRSEL's databases will be incorporated into the web service. This could prove useful for debugging purposes, where, for example, an algorithm fails on a single file.

3.2. Security

Although the D2KWS contains many safeguards, the submission of malicious code, whether intentional or not, must be guarded against. In addition, the protection of copyrighted content against theft must be addressed.

3.3. Language and Platform Support

Supporting multiple OS platforms and programming languages poses significant problems. In the earliest phases of the web service, it is likely that LINUX based MATLAB and Java submissions will be favored, as remotely compiling binaries or supporting precompiled binaries is difficult.

4. FUTURE WORK

Expanding platform and programming language support will be top priority. The likely approach will be to set up a sandboxed machine with identical architecture to the DIY cluster for users to compile their algorithms. In addition, means to gather user feedback for improvement of the system should be implemented. Ways to distribute web service capabilities to other locations outside of IMIRSEL will be explored.

5. ACKNOWLEDGMENTS

This project is funded by the National Science Foundation (IIS-0327371) and The Mellon Foundation.

6. REFERENCES

[1] Downie, J. S. "The Music Information Retrieval Evaluation eXchange (MIREX)," *D-Lib Magazine*, 12 (12), 2006.

[2] Downie, J. S., Ehmann, A. F., Tcheng, D. K. "Music-to-Knowledge (M2K): A prototyping and evaluation environment for music information retrieval research," *SIGIR*, 676, 2005.

[3] Shirk, A. "D2K Web Service Design & Implementation," presented at NCSA CyberArchitecture Working Group, Available at http://algdocs.ncsa.uiuc.edu/PR-20040828-1.ppt, 2004.