MICO — Towards Contextual Media Analysis

Sergio Fernández
Salzburg Research, Austria
sergio.fernandez
@salzburgresearch.at

Sebastian Schaffert
The Apache Software
Foundation
sschaffert@apache.org

Thomas Kurz
Salzburg Research, Austria
thomas.kurz
@salzburgresearch.at

ABSTRACT

With the tremendous increase in multimedia content on the Web and in corporate intranets, discovering hidden meaning in raw multimedia is becoming one of the biggest challenges. Analysing multimedia content is still in its infancy, requires expert knowledge, and the few available products are associated with excessive price tags, while still not delivering sufficient quality for many tasks. This makes it hard, especially for small and medium-size enterprises, to make use of this technology. In addition analysis components typically operate in isolation and do not consider the context (e.g. embedding text) of a media resource. This paper presents how MICO tries to address these problems by providing an open source service platform, that allows to analyse *media in context* and includes various analysis engines for video, images, audio, text, link structure and metadata.

1. THE MICO VISION

The amount of multimedia content on the Web has increased almost exponentially during the last decade, due to cheap media production equipment, everytime-everywhere internet and social media platforms. The pure mass of this data as well as the hidden semantics of raw multimedia content makes it hard to retrieve media assets that satisfy certain information needs. Since common methods for indexing textual content do not fulfill the requirements for multimedia data, there is a need for new methods and technologies. Given the fact, that most of the multimedia content is integrated in so called '*information units*' (spatially related or linked bundles of diverse content formats that are combined to illustrate a certain topic, event or fact), the MICO (Media In Context) project makes use of all surrounding information to enrich the pure content, align existing and new metadata into a common model and provide access methodologies for the emerging cross-media data.

MICO targets a platform that supports the interaction of various multimedia analysis components in a loosely coupled cluster environment. Therefore, we propose a concrete set of harmonized models and software services for orchestration of analysis components, representing and publishing of analysis results. Furthermore we elaborate techniques for multimedia information retrieval including text, image, audio, video, and office documents, in order to provide straight forward methods for accessing and recommending media assets and data.

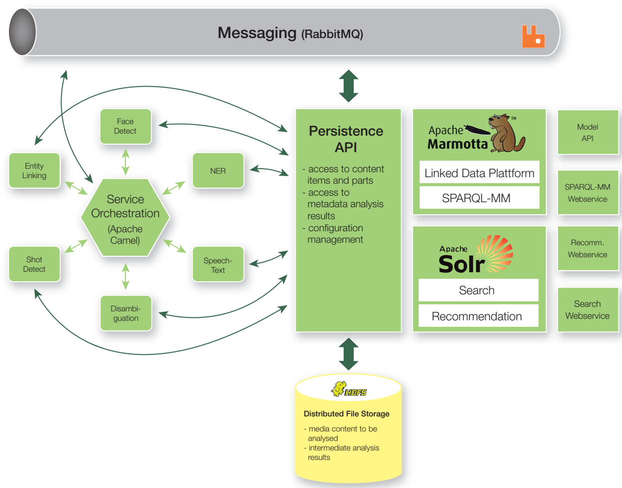

Figure 1: MICO high-level architecture.

2. MICO ARCHITECTURAL APPROACH

The MICO Platform[1] is an environment that allows to break up the hidden semantics of *media in context* by orchestrating a set of different components that collaboratively analyse content, each adding its bit of additional information to the final result. Analysis components can be for instance a *language detector* (identifying the language of a text or audio track), a *keyframe detection* (identifying relevant images from a video), a *face detector* (identifying objects that could be faces), a *face recognizer* (assigning faces to concrete persons), an *entity linker* (assigning objects to concrete entities) or a *disambiguation component* (resolving possible alternatives be choosing the more likely given the context). In case necessary, a more detailed description can be read from the technical deliverables of the project [5, 6].

MICO uses a Service-Oriented Architecture (SOA), where

[1] `http://www.mico-project.eu/platform`

WWW 2015 Companion, May 18–22, 2015, Florence, Italy.
ACM 978-1-4503-3473-0/15/05.
http://dx.doi.org/10.1145/2740908.2742916

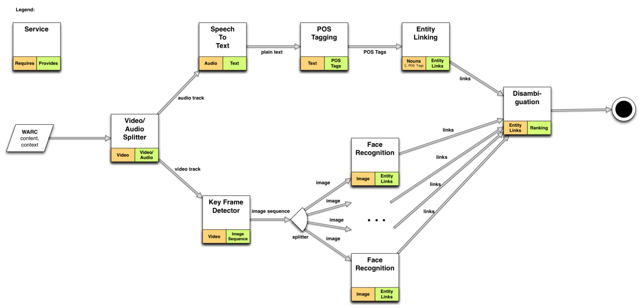

Figure 2: MICO execution plan (example).

each analysis process (extractor) is a component managed by the platform. The Figure 1 depicts this high-level architecture, where an orchestration service plays a crucial role. An example of how an execution plan looks like is shown by Figure 2. The overall process is then supported by some other required components, such a messaging queue for communication, a triplestore for metadata storing and a unstructured data store to persist the raw media files. Further implementation details are provided in the following section. On top of this platform users can build custom solutions for specific use cases.

3. IMPLEMENTATION

As described in the previous section, the MICO Platform follows the SOA paradigm, taking it further beyond. MICO is implemented by moving this paradigm down to the level of the operative system. In MICO each extractor is an independent process, without any restrictions about its resources, neither physical location nor runtime environment. For supporting the analysis chain explained above, each extractor just needs to get registered via a messaging queue (RabbitMQ[2]). Content that is pushed to the platform for analysis purposes is stored in a Hadoop Distributed File System (HDFS[3]). The storage process triggers a dynamically orchestrated process across the different extractors. MICO uses Apache Camel[4] for the implementation of the orchestration, which follows well-known Enterprise Integration Patterns. All metadata results that are produced by the orchestrated analyzers (intermediate as well as final results) are using the MICO RDF vocabulary[5] which can be produced directly by the analysis service or via an API[6]. The API is currently available for Java and C++, the two languages our requirements analysis revealed as the ones which most of our target extractors are available, but it could be easily implemented in any other programming language. All produced metadata is stored in a RDF triple store.

Behind the scenes

The MICO Platform uses mainly Semantic Web technologies and Open Standards. The platform is implemented on top of Apache Marmotta[7] for metadata management. The metadadata is described using a formal RDF vocabulary based in the Ontology for Media Resources [3] and the Open Annotation Model [4]. To link annotations to spatio-temporal content parts, the model uses Media Fragments URIs [8]. Most of the protocols are based on REST, Linked Data Platform [7] and SPARQL [1]. Additionally it provides a native implementation of SPARQL-MM [2], a functional extension for multimedia specific queries.

The source code of the MICO Platform is available from Bitbucket[8], under Apache License Version 2.0.

4. CONCLUSIONS

Although the MICO is still in a early phase, the foundations of the platform have been in place early enough to focus on addressing the actual challenges: effective and productive automatic analysis from media content, combination cross-media analysis results, data provenance, etc. We aim to address all of them in the following two years which include many challenges, technology as well as research-wise, like how to move computation to where the data is stored in a distributed environment and not the other way around, how to build more dynamic and effective execution plans to orchestrate the analysis process depending on dynamic outputs, etc. This paper just presents the very initial outcomes of this project to the relevant scientific community, mainly looking for discussion and valuable feedback to address all those challenges.

Acknowledgments

This work has been partially funded by the European Commission 7th Framework Program, under grant agreement no. 610480.

5. REFERENCES

[1] L. Feigenbaum, G. T. Williams, K. G. Clark, and E. Torres. SPARQL 1.1 Protocol. Recommendation, W3C, March 2013.

[2] T. Kurz, S. Schaffert, K. Schlegel, F. Stegmaier, and H. Kosch. SPARQL-MM, Extending SPARQL to Media Fragments. In *The Semantic Web: ESWC 2014 Satellite Events*, 2014.

[3] W. Lee, W. Bailer, T. Burger, P.-A. Champin, J.-P. Evain, V. Malaise, T. Michel, F. Sasaki, J. Soderberg, F. Stegmaier, and J. Strassner. Ontology for Media Resources 1.0. Recommendation, W3C, February 2012.

[4] R. Sanderson, P. Ciccarese, and H. V. de Sompel. Open Annotation Data Model. Community Draft, Open Annotation Collaboration, Feb. 2013.

[5] S. Schaffert and S. Fernández. D6.1.1 MICO System Architecture and Development Guide. Deliverable, MICO, 2014.

[6] S. Schaffert and S. Fernández. D6.2.1 MICO Platform Initial Version. Deliverable, MICO, October 2014.

[7] S. Speicher, J. Arwe, and A. Malhotra. Linked Data Platform 1.0. Proposed Recommendation, W3C, December 2014.

[8] R. Troncy, E. Mannens, S. Pfeiffer, and D. V. Deursen. Media Fragments URI 1.0 (basic). Recommendation, W3C, September 2012.

[2]`http://www.rabbitmq.com/`

[3]`http://hadoop.apache.org/hdfs/index.html`

[4]`http://camel.apache.org/`

[5]`http://www.mico-project.eu/ns/platform/1.0/schema#`

[6]`http://mico-project.bitbucket.org/api`

[7]`http://marmotta.apache.org/`

[8]`http://code.mico-project.eu/platform`