



**HAL**  
open science

## A Web Services Orchestration Solution for Semantic Multimedia Indexing and Retrieval

Mihaela Brut, Florence Sèdes, Ana-Maria Manzat

► **To cite this version:**

Mihaela Brut, Florence Sèdes, Ana-Maria Manzat. A Web Services Orchestration Solution for Semantic Multimedia Indexing and Retrieval. International Conference on Complex, Intelligent and Software Intensive Systems (CISIS 2009), Mar 2009, Fukuoka, Japan. pp.1187–1192, 10.1109/CISIS.2009.180 . hal-03765447

**HAL Id: hal-03765447**

**<https://hal.science/hal-03765447v1>**

Submitted on 7 Sep 2022

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# A Web Services Orchestration Solution for Semantic Multimedia Indexing and Retrieval

Mihaela Brut, Florence Sedes, Ana Maria Mânzat

**Abstract**— In this article we are presenting a solution for the problem of combining various indexation algorithms in order to acquire a semantic multimedia indexation and to provide responses to the user complex queries. The challenge of this problem concerns the big heterogeneity of the multimedia indexation algorithms and the weak semantic aspect they address. Our solution considers a generic interface for the indexation algorithms, an implementation as Web services, as well as a semantic description in terms of WSMO (Web Service Modeling Ontology) of their functionality and orchestration. Original contribution of the article concerns the idea of organizing the various multimedia metadata types into a generic structure, used to express the user queries, the algorithms' generic interface, as well as the algorithms' WSMO metadata. This approach facilitates the definition of algorithm combination rules, and enables the reduction of the multimedia retrieval task to a metadata matching process.

**Index Terms**— Multimedia indexing, Semantic Web services, algorithm generic interface

## I. INTRODUCTION

VARIOUS domains such as news gathering, TV, banks of resources for commercial or consumer applications, collaborative work, video surveillance were flooded in the last years by a huge amount of video and multimedia sources; now, these domains are in a growing demand of solutions for their management. In this context, LINDO (Large scale distributed INDEXation of multimedia Objects-<http://www.lindo-itea.eu>) ITEA 2-06011 project aims to develop an distributed generic architecture where storage is distributed and where, rather than moving content massively to central processing facilities, the relevant indexation routines are sent to the remote sites to be run locally in each storage environment. Then, only the minimum required content, extracted from its clip or collection, can be transferred as required.

The most sensitive aspect of multimedia distributed objects management concerns the multimedia content semantics: the indexation algorithms for images, audio and video content are mainly in charge with low-level multimedia features analysis.

Mihaela Brut, Alexandru Ioan Cuza University, 11 Carol I Blv., Iasi, Romania, (phone: +4-0746-054774; fax: +4-0232-201490; e-mail: [mihaela@info.uaic.ro](mailto:mihaela@info.uaic.ro)).

Florence Sedes and Ana Maria Mânzat, Institut de Recherche en Informatique de Toulouse, Paul Sabatier University, 108 Route de Narbonne, Toulouse, France (phone: +33-(0)561 557443; fax: +33-(0)561-556258; e-mail: [florence.sedes@irit.fr](mailto:florence.sedes@irit.fr), [ana-maria.manzat@irit.fr](mailto:ana-maria.manzat@irit.fr))

However, the combination of multiple indexation algorithms could lead to semantic information about the multimedia content.

This paper presents a solution, developed inside the Lindo project team, for the problem of combining various indexation algorithms in order to acquire a semantic multimedia indexation and to provide responses to the user complex queries. Because the multimedia indexation algorithms are quite heterogeneous, our solution defines a generic interface for them. In order to enhance the weak semantic they address (being in charge especially with multimedia physical features), our solution accompanies the algorithms by a WSMO semantic description of their functionality and orchestration. The solution is based on a generic metadata structure defined and used to express the algorithms' interface and metadata, as well as the user queries. This approach enables the reduction of the multimedia retrieval task to a metadata matching process.

The paper will present first the generic metadata structure we mentioned above. Then, the definition of a generic interface for the indexation algorithms will be presented, where the input and output data are expressed in terms of the defined metadata structure. The algorithms' semantic description with the support of WSMO will be presented further, emphasizing the facility of specifying algorithm combination rules. An architectural overview of the multimedia indexation and retrieval process will be exposed further. In the end, conclusions and further work directions will be emphasized.

## II. MULTIMEDIA METADATA ORGANIZATION

The metadata provided by the indexation algorithms are expressed into a wide variety of formats, standards, or vocabularies. Many times, a similar metadata is expressed in different formats if it is returned by different algorithms. Moreover, some algorithms return results in form of numerical, Boolean or string values; in these cases, the name and the value of the metadata expressed through such a result is not transparent.

In order to gain a uniform representation of the metadata produced by the indexation algorithms, we conceived a generic metadata structure, where multimedia features are organized in two levels, and original metadata namespaces are managed into a metadata dictionary. Because different metadata vocabularies for a certain multimedia type include some common metadata (under the same or different names)

which specify the same multimedia feature, we selected certain vocabularies for expressing each particular multimedia feature. However, when a new feature is captured by an algorithm, the namespace of the corresponding vocabulary will accompany the metadata.

The mentioned two levels are:

- 1) General metadata, valid for all multimedia types: filename, location, title, subject, author, producer, copyright, keywords, creationDate, GPSPosition, size;
- 2) Media-specific metadata:
  - The text metadata structure contains all the metadata from the DublinCore (<http://dublincore.org/>) element set, some information that can be extracted from the PDF, OpenOffice and MicrosoftOffice documents. Among these elements we could mention language, charCount, wordCount, pages, document etc.
  - The image metadata structure includes the Dublin Core Element Set, the elements of Exif (Exchangeable Image File Format - <http://www.exif.org/>) and IPTC (International Press Telecommunication Council - <http://www.iptc.org/>) standards. For the medical images, the DICOM (Digital Imaging and Communication in Medicine - <http://medical.nema.org/>) format is also added in the structure. Such particular elements are histogramColor, texture, resolution, positionX, positionY etc.
  - The video metadata are expressed through MXF (SMPTE Material eXchange Format - <http://www.smpte-mxf.org/>) standard and include support for video segmentation and object recognition. We could mention segmentVideo, GPS, PTZ, dimension1, dimension2 elements.
  - The audio metadata provide support for expressing audio segmentation (through the MXF/Exif standard). SegmentAudio, topic, speaker, episode, scribe, elapsedTime, and unit are some audio specific metadata elements.

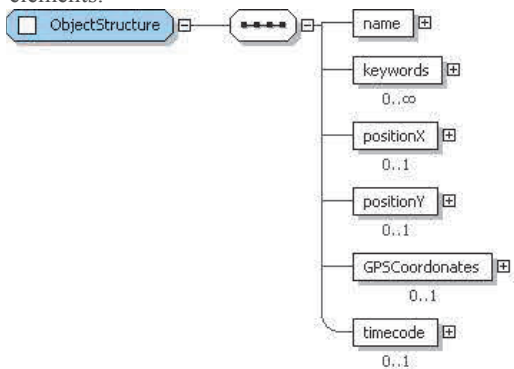


Figure 1: The structure of the metadata describing the multimedia content

In order to express the semantic information about the multimedia indexed content, we included the <object> element inside the metadata structure of each media type. For us an object is any particular semantic information that can be extracted from a media. For this object the structure proposed is presented in Figure 1.

### III. A GENERIC INTERFACE FOR THE MULTIMEDIA INDEXATION ALGORITHMS

The available algorithms for multimedia documents indexing are characterized by a great heterogeneity concerning their input and output data, as well as implementation details, hosting platforms or architectures. In order to uniformly handling these algorithms such as to produce concrete metadata organized according to the above mentioned structure, the algorithms should be structured into a generic interface, where the input and output data are expressed accordingly.

We discuss below some existing approaches in defining generic interfaces and combination rules for the indexation algorithms. In the end of this section, we will present our solution for the generic interface.

Two approaches oriented towards defining such a generic interface are the following:

- 1) The COMM project (<http://comm.semanticweb.org>) proposes a Core Ontology for Multimedia enabling to describe the multimedia types and their particular features. In addition, some patterns for indexing operations are considered, including the semantic annotation pattern and the indexation algorithm pattern, which propose such generic interface.

The multimedia indexation algorithms are considered as Web services, and their results are represented in MPEG-7 format [1]. However, because the lack of a formal semantics of MPEG-7 language, some semantically equivalent results could be represented through some syntactically different MPEG-7 descriptors. The COMM project developed the Core Ontology for Multimedia in order to provide a formal semantics to MPEG-7. Their framework relies on domain-specific ontologies for representing the real world entities depicted in multimedia content.

Our approach differs from the COMM project goal in two aspects: we do not limit the indexation algorithms to a result in MPEG-7 format (our generic metadata structure integrate all kinds of multimedia formats, managed through a metadata dictionary), but we try to limit their scope to detection of perceivable content of the multimedia objects (instead of a more accurate semantic description through domain-specific ontology concepts). The reason of this limitation concerns our desire to exploit the semantics detected by the algorithms themselves, without considering a manual ontology-based annotation.

- 2) The Web Consortium's Algorithm representation use case [5], which proposes algorithm ontology to record and uniformly describe available algorithms for image analysis.

The support for describing the conditions for the algorithm combination is integrated into the generic interface itself: the algorithm ontology includes relations for describing the algorithms input and output, but also relations for describing the preconditions to be fulfilled for algorithm execution and the effects of their execution. Among the possible applications of this algorithm representation, the composition of web

services to automatically analyze media based on user goals and preferences is considered.

As could be noticed, the both approaches are focused on extending the black box of the indexation algorithms with a well defined structure and semantic of the algorithms' input and output data, as well as of their functionality.

In order to separate the description of the input/output data of the indexation algorithm from the semantic description of its functionality and orchestration rules, we include the former inside the generic interface, while the second will be expressed in terms of WSMO.

The indexation algorithm *generic interface* is characterized by:

- *input*: concerns a particular multimedia object. Because the general metadata structure presented in the section II includes all the necessary information to locate and define a multimedia object, we represent the algorithm input through this structure.
- *output*: could be a multimedia object, but also a numeric, Boolean or string value. For representing this output, we adopt the metadata elements describing the indexation result, as media-specific metadata, according the format also presented in the section II. Concrete examples will be provided in the next section.

This algorithm interface is generic in the sense it is able to support any kind of indexation algorithm because the input and output data of such algorithm could be mapped into the described format:

- for any multimedia *input* object, the incorporated metadata could be extracted and organized into the generic structure, it is just a format conversion matter;
- the conversion of *output* data into the proposed metadata structure is a more complicated issue, addressed for the moment through human intervention: the semantics of a numeric, Boolean or string result is not transparent to the machine, but such result illustrates a value for a certain multimedia metadata. As consequence, when a new algorithm is re-written according the generic interface, the coder is in charge with expressing the possible results through corresponding values for a certain media-specific metadata.

Concerning the *implementation* of the algorithms with such generic interface, the presented best practices recommends the Web services approach to be good solution due to the Web services main characteristics: autonomy, composability, choreography, discoverability [2]. Our solution extends this approach with a semantic description of the algorithms expressed through WSMO. We benefit by *WSMO Studio* (<http://www.wsmostudio.org/>) which provides support for the services implementation, as well as compatibility with the framework for algorithms' functionality semantic description and orchestration rules.

## IV. A SEMANTIC ORIENTED SOLUTION FOR INDEXING ALGORITHMS COMBINATION

### A. Combining Multiple Indexation Algorithms

Usually, a user complex query involves more than a single algorithm to be combined and sequentially run over the multimedia content. In the multimedia-indexing domain, the indexation process for responding to such complex queries is produced by applying a sequence of indexation algorithms in a given order. In order to obtain proper algorithms sequential combination, a set of rules should be established, defining conditions for each algorithm execution [3].

We consider as concrete example the following user complex query: "locate the video files where Tim Berners Lee speaks about the semantic Web". In order to build query results, the combination of multiple algorithms is necessary such the recognition of Tim Berners Lee, the English Language detection, specific words recognition, etc. Some rules applicable for this example include: words recognition could be applied if the English Language detection was previously performed; also, the English Language detection could be performed only after the human presence detection, e.g. through the recognition of Tim Berners Lee. These rules could be specified as preconditions specific to each algorithm. In order to establish a algorithm sequential combination for a certain user query, a compatibility checking should be performed based on the corresponding algorithm preconditions.

For expressing such preconditions, we will accompany the indexation algorithms by their semantic description provided with the support of *WSMO*. As example, we will consider the necessity of combining some algorithms exactly in the required sequence.

For the above user query example, we provide below a short description of the *output* data for each algorithm to be called (the input data has always the same structure, as already presented):

- a) Tim Berners Lee presence detection: a successful detection will return an <object> element including the temporal information:

```
<object>
  <name>man</name>
  <keywords>Tim Berners Lee</keywords>
  <timeStamp begin="10:33:00" end="11:33:00" /> ...
</object>
```

- b) The English Language Detection will be applied for the video sequences successfully detected by the previous algorithm and will locate the sub-sequences where English language is present in the audio part. The structure of the returned metadata follows:

```
<audio_metadata lang="en">
  <Speaker id="sp1" name="Tim Berners Lee" type="male"/>
  <timeStamp begin="10:40:00" end="10:50:00" /> ...
</audio_metadata>
```

- c) Recognition of the "semantic Web" phrases will be also applied over the sub-sequences previously located, and

the returned metadata will further specify the relevant sub-sub-sequences:

```
<object>
  <name>word</name>
  <keywords>semantic web</keywords>
  <timeStamp begin="10:44:00" end="10:46:00" /> ...
</object>
```

In a further section, we describe the WSMO solution for locating and combining in such sequence the corresponding algorithms.

For selecting the suitable algorithms to be invoked in order to respond to a user query, we assume the query is pre-processed (e.g. by a *Querying* module, based on natural language processing techniques) and provided into a metadata structured manner. The above query should be received in the following form:

```
<object>
  <name>man</name>
  <keywords>Tim Berners Lee</keywords>
  ...
</object>
<audio_metadata lang="en">
  <Speaker id="sp1" name="Tim Berners Lee" type="male"/>
</audio_metadata>
<object>
  <name>word</name>
  <keywords>semantic web</keywords>
  ...
</object>
```

### B. Existing Approaches for Web Services Semantic Description

Considering multimedia indexation algorithms implemented as Web services (as the best practices recommend) and modeled with the generic interface described in the previous section, we discuss and present further a solution to combine various algorithms by assigning them a semantic description.

Usually, the Web services are located, invoked and combined inside a certain application according the statements established by the application's human coder. He/she assumes the task of Web services *orchestration*, specifying how each service achieves its capability by making use of other Web services.

In the context of the Semantic Web age, some approaches aim to make more machine-processable the Web content accessible through Web services usage. They are focused on assigning semantic annotations to Web services and on exploiting these in order to automate the tasks of Web services discovery, composition and invocation. Thus, the Web services could be integrated inside multiple, distributed applications.

The *OWL Services Coalition* () adopts the vocabulary defined by OWL-S (OWL-based Web Service Ontology) in order to provide semantic annotations of services. An

insufficiency of OWL-S from our problem point of view concerns the lack of support for describing the conditions for combining a set of Web services.

*METEOR-S* (<http://lstdis.cs.uga.edu/Projects/METEOR-S/>) is a technology centered approach, aiming to integrate the multiple Web services technologies, but it does not provide a conceptual model for Web services description.

*IRS-II (Internet Reasoning Service)* is a framework for semantically describing and for executing Web services (<http://kmi.open.ac.uk/technologies/irs/>). With common roots as IRS-II and more focused on the semantic description of Web services, *WSMO* provides the conceptual underpinning and a formal language for semantically describing all relevant aspects of Web services accessible through a Web service interface in order to facilitate the automatization of discovering, combining and invoking services over the Web.

We adopt WSMO because it provides support for all issues related to our problem, especially regarding the usage of Web service semantic description in order to check the possibility of multimedia indexing algorithm combination.

### C. Using WSMO for Web Services Semantic Description and Orchestration

WSMO (*Web Service Modeling Ontology*) is structured in four main elements [4]:

- *ontologies*, providing the terminology used by other WSMO elements;
- *Web services*, providing access to services that, in turn, provide some value in some domain;
- *goals*, representing user desires;
- *mediators*, dealing with interoperability problems between different WSMO elements.

WSMO also provides a logical language for defining formal statements in WSMO.

We are focusing on WSMO support for describing Web services, which enables to depict the functionality of the Web services (through class *Capability*), as well as the orchestration of the Web service (through one or more *Interfaces*).

The *capability* of a Web service is expressed by the state of the world before the Web service is executed and the state of the world after successful Web service provision, and is defined by the following class:

```
Class capability
  hasNonFunctionalProperty type nonFunctionalProperty
  importsOntology type ontology
  usesMediator type oMediator
  hasSharedVariables type sharedVariables
  hasPrecondition type axiom
  hasAssumption type axiom
  hasPostcondition type axiom
  hasEffect type axiom
```

The set of *non-functional properties* are mainly used to describe non-functional aspects of the Web service such as creator, creation date, natural language descriptions, etc.



*Imported Ontologies* allow a modular approach for ontology design when a more formal semantic description is intended for a particular problem domain. Some specific *mediators* are used when an alignment of the imported ontologies is necessary.

*Shared Variables* represent the variables used to define the preconditions, postconditions, assumptions and effects: the accomplishment of the preconditions and assumptions implies the postconditions and effects production. For our problem, each multimedia metadata considered in the two levels generic metadata structure constitutes a shared variable. As example for the above presented three output examples, the following shared variables could be considered: ?object.name, ?object.keywords, ?object.timestamp[begin], ?object.timestamp[end], ?audio\_metadata[lang], ?audio\_metadata.Speaker[id], ?audio\_metadata.Speaker[name], ?audio\_metadata.Speaker[type], ?audio\_metadata.timestamp[begin], ?audio\_metadata.timestamp[end].

*Preconditions* specify what information a Web service requires in order to provide its value. As preconditions we will specify the particular metadata a multimedia object requires in order to be processed by the current indexation algorithm. If this particular metadata is not available, then the particular algorithm(s) to produce it will be located and will be executed first.

We provide below the preconditions for expressing the rules for our particular example. The first precondition concern the English Language detection algorithm, which could be performed only after the human presence detection, e.g. through the recognition of Tim Berners Lee:

precondition

```
axiom preconditionEnglishDetection
definedBy
exists ?name, ?keywords
(?object[name hasValue ?name,
keywords hasValue ?keywords]
memberOf object
and
(?object.name="man" and
?object.keywords="Tim Berners Lee"))
```

The second precondition concerns the words recognition algorithm, which could be applied if the English Language detection was previously performed:

precondition

```
axiom preconditionWordRecognition
definedBy
exists ?lang, ?name
(?audio_metadata[
lang hasValue ?lang,
Speaker[name] hasValue ?name]
memberOf audio_metadata
```

and

```
(?audio_metadata[lang]="en" and
?audio_metadata.Speaker[name]="Tim Berners Lee")
```

Preconditions' specification is enough for our particular problem (suitably combining indexation algorithms). However, the WSMO specification provides also support for expressing:

- *Assumptions*: describe the state of the world which is assumed before the execution of the Web service, but is not necessarily checked by the Web service. For our example, we could assume a video file with Tim Berners Lee speaking about the Semantic Web exists, is correct located and could be properly processed.
- *Postconditions*: describe the state of the information space that is guaranteed to be reached after the successful execution of the Web service.

The postcondition should not reproduce the algorithm's output format and content, but just its essential information, eventually related to the input information. For the first example, the postcondition could (facultatively) emphasize the name of the speaker, which should be the same as those mentioned in the precondition:

postcondition

```
axiom postconditionEnglishDetection
definedBy
exists ?name,
(?name memberOf audio_metadata.Speaker[name] and
?name = ?object.keywords)
```

- *Effects*: should be mentioned, for example, when the multimedia object suffers an alteration, such as the image color segmentation, or video color enhancement.

#### D. The Proposed Service Oriented Architecture for Multimedia Indexing and Retrieval

As could be noticed from the previous discussion, in order to accomplish a complex *multimedia indexation task*, three main collections should be handled:

- *multimedia collection*: when a new object is added to this, the general metadata are extracted (by applying a set of implicit indexation algorithms, denoted as "implicit extractors" in Figure 2) and included inside the metadata collection;
- *metadata collection*: metadata associated with each multimedia object is structured on the two levels (general and media-specific). The first level is obtained as described above, while the second is enriched with every new object indexation process;
- *indexation algorithms collection*: when a new algorithm is added, it is processed in two ways:
  - it is re-written as a Web service where the input and output data are converted to the generic interface;
  - its combination rules are specified through WSMO *Capability* class.

Because the *user queries* could be reduced to the retrieval of multimedia objects having some particular features, ideally this queries should be expressed through the metadata illustrating these features, as in the example provided in Section IV.A. We suppose this ideal case, where the query conversion task belongs to a *querying module*, separated by the *indexation module* focused by our discussion.

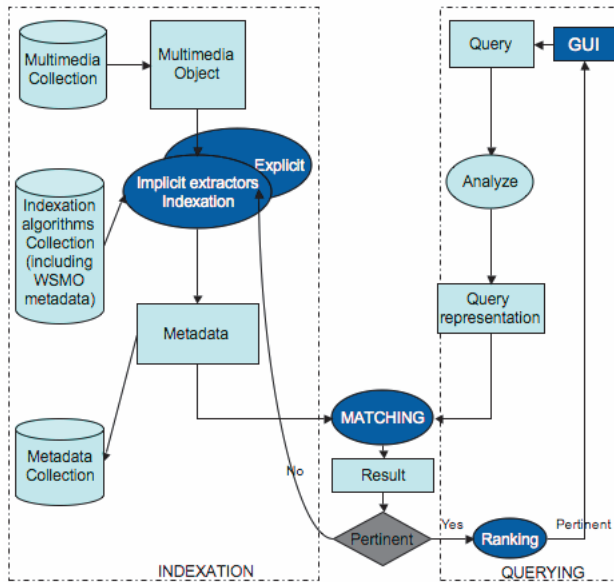


Figure 2: The service-oriented architecture for multimedia indexing and retrieval

For responding to the user queries, a *matching* process between the content representation (the metadata format) and the query representation (available in the same metadata format) should be accomplished. The querying module should perform the results ranking and should transmit if the results are or not acceptable (pertinent). If the results are not pertinent, the indexation module should accomplish a new multimedia objects indexation, by using additional algorithms. This process consists in some steps:

- Analyze the query representation in order to establish the list of additional indexation algorithms (denoted as “explicit extractors” in Figure 2) necessary to be applied. The output format of these algorithms should include metadata specified by query.
- Consult the WSMO metadata associated with these algorithms in order to establish the cascade for the previous extractors list.
- Analyze the query representation in order to establish the multimedia objects sub-collection possibly considered by the user query;
- Index this collection based on the previous established extractors cascade.

In Figure 2, the multimedia indexation process is illustrated, as well as the matching process performed in order to respond to a user query. All the tasks represented through circle forms are conceived as Web services in this distributed architecture.

## V. CONCLUSIONS AND FURTHER WORK

This paper presents a solution for the problem of combining various indexation algorithms in order to acquire a semantic multimedia indexation and to provide responses to the user complex queries. Because the multimedia indexation algorithms are quite heterogeneous, our solution defines a generic interface for them. Because the indexation algorithms are in charge especially with multimedia physical features detection and analysis, our solution try to enhance the weak semantic they address, accompanying the algorithms by a WSMO semantic description of their functionality and orchestration. As original contribution, the solution is based on a generic metadata structure defined and used to express the algorithms’ interface and WSMO metadata, as well as the user queries. This approach enables the reduction of the multimedia retrieval task to a metadata matching process. Service oriented architecture is provided in order to illustrate the proposed solution.

As further work, we intent to adopt and integrate the proposed solution into the distributed multimedia environment developed by the Lindo project, where multimedia indexation and retrieval is coordinated from a central server, but is effectively performed at the level of local nodes, where various types of multimedia collections are located.

## REFERENCES

- [1] Arndt, R., Troncy, R., Staab, S., Hardman, L. Vacura, M., “COMM: Designing a Well-Founded Multimedia Ontology for the Web”, in K. Aberer et al. (Eds.): *Proceedings of ISWC/ASWC 2007*, LNCS 4825, pp. 30–43, Springer, 2007
- [2] Erl, T., *Service-Oriented Architecture: Concepts, Technology, and Design*, Prentice Hall PTR, 2005
- [3] Haidar, B., Joly, P., Haidar S., “A Graph-based Approach to Automatically Chain Distributed Multimedia Indexing Services”, in *Proceedings of 9th Int. Conf. on Internet and Multimedia Systems and Applications*, ACTA Press, Switzerland, 2005;
- [4] Roman, D., Keller, U., Lausen, H., de Bruijna, J., Lara, R., Stollberg, M., Polleres, A., Feier, C., Bussler, C., Fensel, D., “Web Service Modeling Ontology”, *Applied Ontology 1* (2005) pp. 77–106, IOS Press, 2005
- [5] Tzouvaras, V., Troncy, R., Pan, J., “Multimedia Annotation Interoperability Framework”, *W3C Technical Reports*, Boston, 2007: <http://www.w3.org/2005/Incubator/mmsm/XGR-interoperability/>