

# The Rhizomer Semantic Content Management System

Roberto García<sup>1</sup>, Juan Manuel Gimeno<sup>1</sup>, Ferran Perdrix<sup>1,2</sup>, Rosa Gil<sup>1</sup>, Marta Oliva<sup>1</sup>

<sup>1</sup> Universitat of Lleida  
Jaume II, 69  
25001 Lleida, Spain  
[{rgarcia, jmgimeno, ferranp, oliva, rgil}@diei.udl.cat](mailto:{rgarcia, jmgimeno, ferranp, oliva, rgil}@diei.udl.cat)

<sup>2</sup> Segre Media Group  
Del Riu 6  
25007 Lleida, Spain  
[fperdrix@diarisegre.com](mailto:fperdrix@diarisegre.com)

**Abstract.** The Rhizomer platform is a Content Management System (CMS) based on a Resource Oriented Approach (RESTful) and Semantic Web technologies. It achieves a great level of flexibility and provides sophisticated content management services. All content is described using semantic metadata semi-automatically extracted from multimedia content, which enriches the browsing experience and enables semantic queries. A usable user interface is built on top of the CMS in order to facilitate the interaction with content and enhance it with the information provided by the associated semantic metadata. As an application scenario of the platform, its use in a media company where audio content is managed and its speech transcript semantically annotated is described.

## 1. Introduction

Existing CMSs lack consistent and scalable content annotation mechanisms that allow them to deal with the highly heterogeneous domains that information architectures for the knowledge society demand. Recently, some CMS and wiki systems have started to incorporate semantic metadata modules in order to cope with this lack, such as Drupal RDF Modules<sup>1</sup> or the RDF Tools for Wordpress [1]. It is even planned that the next version of Drupal [2] will be entirely based on Semantic Web technologies [3].

However, all these initiatives, except maybe for the future Drupal 7, do not constitute a fully featured semantic CMS, especially if one considers content beyond HTML documents. The same applies to many initiatives from the Semantic Web community. For instance, ODESeW [4] (a Semantic Web application development platform), ontology editors like SWOOP [5], semantic wikis like the semantic extension for Media Wiki [6] (which mix wiki mark-up and semantic annotations) or RDF Browsers like Tabulator [7] or Disco [8] (which just provide browsing capabilities and in some cases metadata edition).

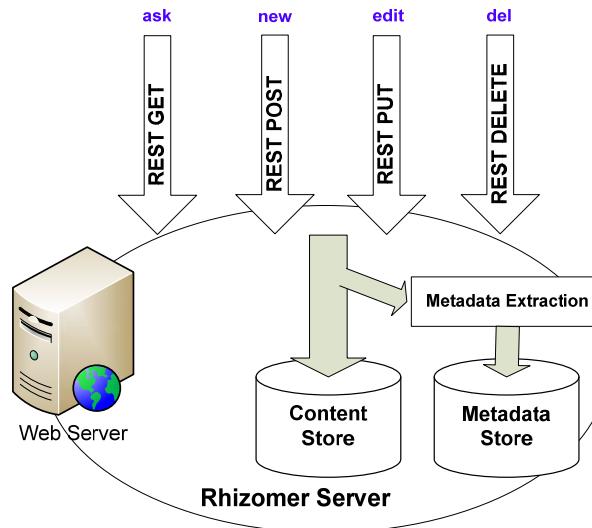
<sup>1</sup> Drupal RDF modules, <http://drupal.org/node/222788>

The aim of the Rhizomer platform is to develop a multimedia CMS combined with a fully featured semantic metadata repository with reasoning capabilities. Both components, the CMS and the semantic repository, are integrated in a transparent way for the end-users and enable more sophisticated and usable interactions.

The Semantic Content Management System (SemCMS) is based on simple foundations, which make it flexible, scalable and capable of adapting to different deployment and use scenarios. Its core is rooted in simple HTTP mechanisms and follows a RESTful approach [9].

Each content item managed by the SemCMS has its own URL, thus basing the whole system on a Resource Oriented Approach. The basic HTTP commands (GET, PUT, POST and DELETE) allow managing each resource. Those same commands are also used to manage semantic metadata about content items: the GET command is employed to pose semantic queries; similarly, PUT, POST and DELETE are used to update resource metadata, generate a new annotation or remove it respectively.

The HTTP functionality is implemented by the Rhizomer server component, shown in Fig. 1. There are additional functionalities, especially those geared towards making the user interaction with the Rhizomer server more usable, which are encapsulated in the user browser and implemented using Javascript and asynchronous HTTP calls (AJAX [10]).



**Fig. 1.** The Rhizomer Server architecture

The Rhizomer platform is described in more detail in Section 2, where its metadata and content managing functionalities are explained. Next, a sample scenario where this platform has been applied is presented in Section 3. Finally, the future plans and the conclusions are outlined in Section 4.

## 2. The Rhizomer Platform

As it has been presented in the introduction, the aim of the Rhizomer platform<sup>2</sup> [11] is to constitute a generic content management system capable of dealing with very heterogeneous content items thanks to its semantic metadata foundations. Moreover, it is based on simple HTTP commands in order to make it more scalable and adaptable to different deployment environments.

HTTP commands are mapped to common data management operations following the REST approach. GET retrieves the content item, PUT updates the specified URL with the provided content, POST creates a new content item in the CMS generating its corresponding URL and DELETE removes the specified item.

The same commands are used for semantic metadata management. However, the GET command for metadata is also used to pose semantic queries based on the SPARQL standard [12]. These queries retrieve both the metadata for a given resource and the descriptions for resources satisfying the constraints defined by the query.

Moreover, in some cases the semantic metadata might refer to resources outside the CMS scope, for instance people or places. These resources have URIs that do not belong to the URL space where the CMS is deployed and therefore cannot be retrieved using the REST approach. In this case, SPARQL queries sent through the GET command are used in order to retrieve metadata.

Therefore, the resources described by semantic metadata can be both content and non-content entities. For non-content resources, HTTP commands operate on the pieces of metadata (RDF triples) describing those resources. For content elements, managed by the Semantic CMS (HTML documents, images, videos...), HTTP commands operate on both the content item and its associated metadata.

Content elements are described using semantic metadata. Whenever a new content is stored in the SemCMS, a metadata extraction plugin specifically tailored for this content type is triggered to extract relevant metadata and store it in semantic form, as it is shown in Fig. 1.

The full user experience is built on top of these operations. However, RDF, the language of the semantic web, is completely hidden in order to increase usability. As end-users typically interact with HTML web pages through their browsers, Rhizomer incorporates a generic transformation from RDF to HTML. This transformation is not tied to any particular ontology or scheme, as it is the case in most template-based approaches. This HTML rendering is used to build a transparent browsing experience on top of the SPARQL endpoint that retrieves semantic metadata.

The browsing steps are based on a fragmentation of the underlying RDF graph, which is detailed in Section 2.1. The same fragments are used in order to constraint the range of the update and deletion operations, as it is detailed in Section 2.2. Updates and new metadata generation are carried out through semantic-enabled HTML forms, which also help to hide the burdens of RDF metadata from users.

Besides, Rhizomer incorporates Semantic Web services providing interaction means beyond content retrieval and metadata browsing. Each user action corresponds to a Semantic Web service whose description incorporates the constraints a resource

<sup>2</sup> <http://rhizomik.net/rhizomer>

must satisfy to be a valid input for the service. Consequently, the semantic description of a resource determines which actions can be applied to it.

For instance, let us consider a scenario where the platform is used to retrieve and browse a set of news items described using semantic metadata. These descriptions include date and time information and, in some cases, the geographical localization of the event covered by the news items. At first, these descriptions are visualised as generic HTML pages based on the RDF to HTML rendering. These pages allow the user to visualise the descriptions of the corresponding resources, the metadata, and to browse them interactively by navigating the underlying graph.

This generic approach can be applied to any kind of resource. However, in this scenario, it would surely be desirable to have more specific views and more accurate ways to interact with events. Calendars or timelines are good choices for time stamped resources, while maps should be helpful for geographically located ones.

The objective of the Rhizomer platform, and the reason why Semantic Web services have been chosen as the way to implement actions, is to build a generic and dynamic system which can directly deal with generic RDF metadata while being easily extensible to incorporate specialised ways to view and interact with particular kinds of resources.

Finally, in order to populate the semantic metadata describing content items, the platform is able to incorporate different metadata extraction plug-ins. Section 2.3 shows some examples of this functionality. It details how metadata can be automatically extracted when new content items are uploaded to the platform. Then, the resulting metadata can be browsed as detailed in Section 2.1 and edited in order to correct or enrich the metadata automatically extracted as detailed in Section 2.2.

## 2.1. Metadata Browsing

Browsing is the basic interaction paradigm in the Web. It is based on the successive visualisation of Web pages following the links connecting them. Pages and links are the main building blocks upon which the interaction is built. Web pages are intended for human users' consumption and well-established methodologies to make them usable and accessible exist.

However, neither the browsing paradigm nor the principles to make the whole thing usable and accessible can be directly applied to the Semantic Web. That is so because it is based on a model not built upon pages and links but on triples (subject-predicate-object), which makes the browsing approach quite different from the Web.

The combination of many triples builds up a graph and, though the resulting model is easier to process by computers, the consumers of Semantic Web metadata are, at the end, humans so usable and accessible interaction mechanisms are required.

First of all, the basic browsing paradigm should change because the Semantic Web makes it very difficult to base the browsing steps on documents. In other words, it does not seem appropriate, for each step, to show all the triples in the corresponding document to the user as it is done in the Web. The amount of information in a single document can be too large, more than thousands of triplets. Moreover, the frontiers among documents are very fuzzy in the Semantic Web: usually, many documents are combined in order to get a coherent graph.

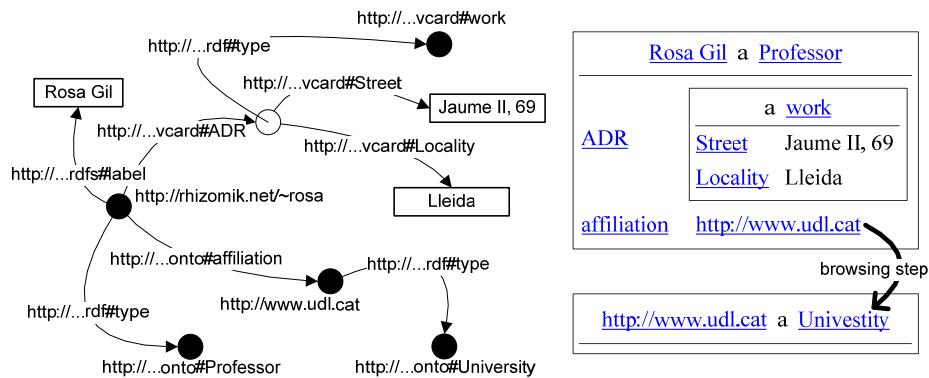
Semantic Web browsers like Tabulator [7] follow this approach and show all the triples from a Semantic Web document as an unfoldable tree. As preliminary user tests show, this approach causes many usability problems because, as the tree grows, it rapidly becomes difficult to manage. As it has been said, documents contain many triples and, additionally, each navigation step adds more triples from the new document to the current set.

Another approach is faceted browsing, as in /facet [13]. However, our objective is a simpler and more polyvalent browsing mechanism, which deals better with heterogeneous information spaces, although it might lack the guidance provided by facets. Anyway, it is not clear how systems like /facet deal with metadata structures that feature many anonymous resources, as it is the case for the semantic metadata managed in the application scenario described in Section 3.

Now, the problem is where to put the limits of each browsing step when presenting semantic metadata. In other words, how each browsing piece is built and how new pieces are created and presented following user needs in order to compose a browsing experience through the whole graph.

In order to facilitate browsing, the proposed approach is based on the construction of graph fragments. Following this approach, it is possible to construct fragments for any graph starting from any node that is not anonymous. For instance, the starting point for the metadata describing a piece of content is the node that represents the piece, which is the subject for all the triples describing it. This node has an ID and consequently is not anonymous.

All the triples that start from this node are part of the fragment and the triples describing anonymous objects are also added to the fragment. This happens for all nodes that are only identifiable in the context of the starting node. For instance, Fig. 2 shows how an example graph would be fragmented following this approach.



**Fig. 2.** Fragmentation of an example RDF graph and the resulting HTML rendering

As it can be seen, there are two fragments, each one corresponding to an identified resource described by at least one triple, for which it is the subject. The first fragment describes <http://rhizomik.net/~rosa> and includes an anonymous resource for the address. The second one, for <http://www.udl.cat>, can be reached from the first one

through a browsing step. Unlike the address, it is shown independently because it is not anonymous.

This process of building fragments continues iteratively and interactively because from a fragment, if the user wants more information about a resource, it is possible to follow the next browsing step starting from any non anonymous object: the metadata describing the selected identified node is retrieved and the new fragment is built. Moreover, when the results are rendered to the users all the URIs are replaced by labels, if they are available, making the results much more usable.

Finally, fragments are rendered using HTML, which is viewable using a web browser, a tool users feel comfortable with. In order to generate HTML from RDF, fragments are serialised as RDF/XML and transformed using an XSL. The XSL transformation, which is part of the Rhizomer platform, guarantees consistent results whenever the input RDF/XML has been generated from fragments based on the Rhizomer approach.

This mechanism has been implemented as successive DESCRIBE queries for the identified resource URIs to the SPARQL endpoint. The DESCRIBE operation of the SPARQL endpoint has been reimplemented in order to build the proposed fragments, which also include all the available labels. Then, the XSL transformation from RDF/XML to HTML is invoked from the client using AJAX, which is also responsible for sending the SPARQL queries and making the whole process go smoothly behind the scenes, making the user experience even more comfortable.

## 2.2. Editing Metadata

The previous fragment-based approach, besides being the foundation for browsing, allows constraining the metadata editing and deletion actions to a limited set of triples. This way, it is possible to implement editing actions as the replacement of a given fragment with the one resulting from the editing process. The same strategy applies for the deletion action.

All these operations are also carried out through an HTML interface. In addition to the RDF to HTML transformation, the Rhizomer platform includes an XSL transformation from RDF to HTML forms. These forms are generated automatically from the RDF/XML corresponding to a fragment. The same approach as in the RDF to HTML transformation is followed but, instead of generating text values and links for literals and resource, this transformation generates input fields for each triple. The field is named using the corresponding property URI and its value corresponds to the triple value. The fields can be used in order to edit the property value, either a resource URIs or a literal.

Moreover, properties and values can be removed or added. Currently, the user enjoys little assistance during the editing process. Basically, when the user chooses to add a new property, a SPARQL query is used in order to retrieve all the available properties for the resource being edited. These are the properties not constrained to a particular resource type plus those constrained to the types of the resource being edited. The future plan is to improve this support in order to assist users during the whole editing process, as it is detailed in Section 4.

Finally, an algorithm has been developed in order to reverse the mapping from RDF to HTML forms. In other words, this algorithm is responsible for generating the RDF that results from the editing process by mapping the form input fields to the corresponding triples. This completes the roundtrip for RDF metadata editing from RDF to HTML forms and back to RDF.

### 2.3. Content Uploading and Annotation

As it has been previously introduced, a side effect of a content item upload is the metadata extraction step. A metadata extraction plug-in specific for the content type is triggered in order to extract relevant metadata and store it in semantic form.

For instance, if an HTML document is stored, the corresponding plug-in is executed in order to generate semantic metadata associated to the document URL. This metadata specifies attributes such as “title”, “keywords”, etc. available from the uploaded document. If the content is a picture, EXIF [14] metadata is extracted and stored as semantic metadata. For video content, automatic metadata extraction applications are used in order to retrieve relevant metadata. The semantic metadata generated is based on semantic web standards.

For editorial metadata (title, date of publication...) Dublin Core is considered and for content metadata (audio and video features, temporal decompositions, structure, knowledge captured by content...) the ontology versions of standards such as MPEG-7 [15], TV-Anytime... are considered. In any case, the metadata extracted is just an annotation proposal. The user can edit it through the user interface previously detailed.

Additional annotations can be triggered for HTML and other textual content. Text content is semantically annotated based on natural language processing techniques. These might be as simple as named entities detection. In this case, when a piece of content is uploaded to Rhizomer, for instance a news item or a blog post, external services are used in order to generate content-based semantic annotations. Two annotation services have been tested so far, the Open Calais API<sup>3</sup> from Reuters and the Freeling [16] Open Source NLP package. Moreover, if content is uploaded from external URLs, a plug-in recovers the tags attached to them in collaborative bookmarking services such as del.icio.us<sup>4</sup> and generates semantic versions of the tags. Textual semantic annotations can be leveraged to audio content too. As it is detailed in Section 3, for speech in audio content a transcript might be generated thus allowing the resulting text to be annotated as any other textual source.

## 3. Application Scenario

The Rhizomer platform has been put into practice in the Segre<sup>5</sup> media group in the context of the S5T<sup>6</sup> research project. S5T extends semantic annotation to the audio

<sup>3</sup> <http://www.opencalais.com>

<sup>4</sup> <http://del.icio.us>

<sup>5</sup> <http://www.diarisegre.com>

part of audiovisual contents. In order to do that, a transcript of the audio voices is automatically generated and processed in order to detect key terms and produce semantic annotations based on these terms. More details about this process are available in [17].

The news items management system is based on the Rhizomer platform and allows browsing the audiovisual contents through their transcripts or, in a complementary way, together with the ontologies and metadata in their semantic annotations. All the metadata and the ontologies are based on RDF, so they can be browsed in a generic way by means of the RDF to HTML transformation provided by the platform. It is also possible to perform queries through the SPARQL endpoint.

Metadata for audio content items is generated following the audio transcript plus semantic annotation process described before. This process has been integrated into the Rhizomer platform as a metadata extraction plug-in. When an audio file corresponding to a news item is uploaded to the Rhizomer platform in the S5T project, the audio transcript is generated and the process continues with a semantic annotation similar to the one carried out by the OpenCalais API commented in Section 2.3. In this case, ontologies tailored to the kind of news items managed in the project are used in order to produce more relevant semantic annotations.

On top of this content and metadata base, some services have been added in order to carry out actions specific to the Segre scenario and which depend on the type of object being manipulated. For audiovisual contents with an annotated transcript, a semantic web service implementing one of these specific actions has been added, which is described in Section 3.1. The rest of the services and features, e.g. metadata rendering and browsing, are directly reused from the Rhizomer platform.

### 3.1. Transcript-based Interaction Service

In the Segre scenario, there is a specific interaction action enabled for audiovisual resources, i.e. resource of type *mpeg7:AudioType*, with an associated *transcript* property. The corresponding web service provides a view, shown in the right part of Fig. 3, which permits additional interaction possibilities through the semantic annotations automatically generated [17] from the audio transcript.

This view permits rendering audio and video content and interacting with it through a clickable version of the audio transcript. Two kinds of interactions are possible from the transcript. First, it is possible to click on any indexed word in the transcript in order to perform a keyword-based query for all the pieces of content containing that keyword. Second, the transcript is enriched with links to the ontology used for semantic annotation. Each word whose meaning is represented by an ontology concept is linked to a description of that concept.

For instance, the transcript includes the name of a politician indexed and modeled in the ontology. Consequently, it can be clicked in order to get all the audiovisual items where his name appears or, alternatively, to browse all the knowledge about that politician encoded in the corresponding domain ontology.

<sup>6</sup> <http://nets.ii.uam.es/~s5t>

<p>...audio/20070113 a <a href="#">AudioType</a></p> <p><a href="#">title</a> Butlletí Nit  <a href="#">date</a> 2007-01-13  <a href="#">genre</a> politics  <a href="#">transcript</a> <a href="http://...0113.xml">http://...0113.xml</a></p> <p style="text-align: right;"><a href="#">play</a></p> <hr/> <p>...audio/20070322 a <a href="#">AudioType</a></p> <p><a href="#">title</a> Butlletí Migdia  <a href="#">date</a> 2007-03-22  <a href="#">genre</a> politics  <a href="#">transcript</a> <a href="http://...0322.xml">http://...0322.xml</a></p> <p style="text-align: right;"><a href="#">play</a></p>	<p><b>http://www.segre.com/audio/20070113.mp3</b></p>  <p><a href="#">Search Keyword</a>  <a href="#">Browse Term</a></p> <p>La mobilització en contra dels <a href="#">transgènics</a> de <a href="#">Josep Pàmies</a> també ha servit per introduir altres reclamacions. En aquest cas, alguns dels col·lectius de la lluita contra aquests cultius demanen que la <a href="#">Universitat de Lleida</a> rebi una especialització en <a href="#">Agricultura Ecològica</a>. Asseguren que serien uns estudis pioners que servirien al centre per recuperar prestigi.</p>
--	--

**Fig. 3.** Metadata view (left) and transcript view (right) available through the "play" service

#### 4. Conclusions and Future Work

The Rhizomer platform constitutes a simple yet flexible content management system. Thanks to its semantic metadata, it facilitates content management in heterogeneous scenarios and builds an improved user experience. Metadata is produced when content items are uploaded to the platform. Specialised metadata extraction plug-ins are incorporated and extract relevant metadata depending on content type, from EXIF metadata for pictures to semantic annotations for text content.

The platform offers a generic RDF to HTML transformation that makes it possible to navigate through semantic metadata and the associate ontologies. Apart from metadata browsing capabilities and content retrieval, users can carry out additional actions implemented by means of Semantic Web services. These services are associated to the resources by a matching process based on their semantic descriptions.

The platform has been applied in the Segre media group to develop an audio news items management system. The platform features a metadata extraction plug-in that produces semantic annotations for the audio voice transcripts of the uploaded audio files. The resulting metadata is used by a specialised interaction service that plays audiovisual content together with a transcript for the audio voice. The transcript is enriched with semantic annotations that can be used to retrieve other pieces of content or browse the semantic annotations metadata.

Future work focuses on metadata edition facilities. In addition to the assistance when a user tries to add a new property to the current description. Property ranges and restrictions that apply to the kind of resource being edited will be considered to suggest resources which could constitute a proper value for the property. Moreover, new metadata extraction plug-ins and interaction services for additional application scenarios will be explored. For instance, plug-ins to extract information from collaborative bookmarking and ranking sites will also be considered, trying to leverage information already present on the web.

## Acknowledgements

The work described in this paper has been partially supported by Spanish Ministry of Science and Education through the Scalable Semantic personalised Search of Spoken and written contents on the Semantic Web (S5T) research project (TIN2005-06885).

## References

1. Nowack, B.: RDF Tools - An RDF Store for WordPress. BNode, January 15th (2008). <http://bnode.org/blog/2008/01/15/rdf-tools-an-rdf-store-for-wordpress>
2. Buytaert, D.: DrupalCon Boston 2008: State of Drupal presentation. DrupalCon'08, Boston, MA (2008). <http://buytaert.net/files/state-of-drupal-2008-boston.pdf>
3. Lytras, M., García, R.: Semantic Web Applications: A framework for industry and business exploitation - What is needed for the adoption of the Semantic Web from the market and industry. International Journal of Knowledge and Learning 4(1), in press (2008)
4. Corcho, O., López-Cima, A., Gómez-Pérez, A.: The ODESeW 2.0 semantic web application framework. In : Proceedings of the 15th International Conference on World Wide Web, WWW '06. ACM Press, pp. 1049-1050 (2006)
5. Kalyanpur, A., Parsia, B., Hendler, J.: A Tool for Working with Web Ontologies. International Journal of Semantic Web and Information Systems 1(1), 36-49 (2005)
6. Krötzsch, M., Vrandečić, D., Völkel, M.: Semantic MediaWiki. In : Proceedings of the Int. Semantic Web Conference, ISWC'06. LNCS vol. 4273, pp. 935-942, Springer (2006)
7. Berners-Lee et. al: Exploring and Analyzing linked dates on the Semantic Web. In : Proc. of the 3rd International Semantic Web User Interaction Workshop (2006)
8. Bojars, U., Passant, A., Giasson, F., Breslin, J. G.: An Architecture to Discover and Query Decentralized RDF Data. In : Proceedings of the ESWC'07 Workshop on Scripting for the Semantic Web, SFSW 2007. CEUR Workshop Proceedings, vol. 248 (2007)
9. Richardson, L., Ruby, S.: Restful Web Services. O'Reilly (2007)
10. Crane, D., Pasarello, E. and James, D.: "Ajax in Action". Manning Publications, 2005
11. García, R., Gil, R.: Improving Human-Semantic Web Interaction: The Rhizomer Experience. CEUR Workshop Proceedings, Vol. 201, pp. 57-64, 2006
12. Prud'hommeaux, E., Seaborne, A. : SPARQL Query Language for RDF. Recommendation, World Wide Web Consortium (2008). <http://www.w3.org/TR/rdf-sparql-query>
13. Hildebrand, M., Ossenbruggen, J., Hardman, L.: /facet: A Browser for Heterogeneous Semantic Web Repositories. In : Proc. of the International Semantic Web Conference 2006. Lecture Notices in Computer Science, Vol. 4273, pp. 272-285 (2006)
14. Kanzaki, M.: Exif vocabulary workspace - RDF Schema. W3C Technical Report (2003). <http://www.w3.org/2003/12/exif>
15. García, R., Tsinaraki, C., Celma, O., Christodoulakis, S.: Multimedia Content Description using Semantic Web Languages. In : Y.Kompatsiaris and P. Hobson (eds.) Semantic Multimedia and Ontologies: Theory and Applications. Springer (2008)
16. Atserias, J., Casas, B., Comelles, E., González, M., Padró, L. and Padró, M.: FreeLing 1.3: Syntactic and semantic services in an open-source NLP library. In : Proc. of the 5th Int. Conf. on Language Resources and Evaluation (LREC'06), Genoa, Italy (2006)
17. Tejedor, J., García, R., Fernández, M., López, F., Perdrix, F., Macías, J.A., Gil, R., Oliva, M., Moya, D., Colás, J., Castells, P.: Ontology-Based Retrieval of Human Speech. In : Proc. of the 6th International Workshop on Web Semantics, WebS'07. IEEE Computer Society Press (2007)