ISSN 2395-8618

Recommendation for an Enterprise Content Management (ECM) Based on Ontological Models

José Márquez, Manuel Escalante, Leonardo Sampedro,
Elba Sánchez, Laura Ortiz, and Eduardo Zurek

Abstract—**This paper presents a system of recommendations for an enterprise content manager (ECM) based on ontological models. In many occasions the results of a search are not accurate enough, so the user of the ECM system must check them and discard those not related to the search. In order to make recommendations, a proposal where it is necessary to review the instances of the ontological model is presented to manage the alias and ambiguities. Comparisons are made between the results obtained from the traditional search model and the recommendations suggested by the model proposed in this work.**

Index Terms—**Ontologies, ECM, natural language processing, searching, recommendations, semantic web.**

I. Introduction

In an Enterprise Content Management (ECM) system, the search is a critical and repetitive task. Access to the requested information is vital for the person who performs the search, but the information is not always presented explicitly, as when the search is done by date and author. For this reason, the person must read the documents and determine if the result is correct or not. In the market, there are different commercial solutions that implement ontological models in an ECM, such as Athento ECM [1] – Zaizi [2], but they do not reveal how to use ontological models because this is a commercial secret. For a content management user, it is important to have all the documents organized and have all the control access for the documents.

This paper describes recommendation system based on ontological models. The models give solution to two of the most common problems: ambiguity and alias, which are handle in order to give the final user some suggestions about other documents that could have any relation with the search terms.

The work described here is part of a research project founded by COLCIENCIAS, the entire project is aimed to the development of a recommendation system for an ECM software.

Two ontological models were applied to represent entities from the content of the documents. The results of applying the FOAF [3] model, with the property TheSameAs, can be used to present to the final user documents that are related with some person but are referenced with a nickname or alias, and cannot be reached with the traditional model of search. The second model has a special property, HasFacet, which enable the instances of the model to have relations with instances of other models such as those that we show here with car [4] and places [5] model.

This paper has seven parts. Section 1 introduces the work presented here. Section 2 describes some issues with the current search technique based on key words. Section 3 shows works that has been done by some ECM companies, and research on using semantic, ontology, and ECM to manage data and information in a different way. Section 4 presents information about ontology. Section 5 describes our proposal to handle ambiguity and alias problems on ECM. In section 6 we present some results after applying our proposal to a search engine, and show the differences with the current search method, and finally in Section 7 we present some conclusion of this work.

II. Problem

The problem we address can be formulated as follows: “Enterprise Content Management (ECM) makes reference to the strategies, methods and tools to capture, manage, storage, preserve and present the contents handled by an organization” [6].

In an ECM, it is not possible to find more relations between the objects that are part of the system, but only those established in the database design. Basically, in an ECM, we can make consultations about documents in a specific status, to consult the name, date or any other metadata, or find a word

Manuscript received on February 02, 2016, accepted for publication on May 24, 2016, published on June 25, 2016.

J. Marquez, E. Zurek, and M. Escalante are with the Universidad del Norte, Km. 5 Autopista a Puerto Colombia, Barranquilla, Colombia (e-mail:{jmarquez, ezurek}@uninorte.edu.co, wolverineun@hotmail.com).

L. Sampedro, E. Sánchez, and L. Ortiz are graduate students in Systems Engineerning, Universidad del Norte, Km. 5 Autopista a Puerto Colombia, Barranquilla, Colombia (e-mail: {ljsampedro, bita.sanchez2991, lauraortizmartinez}@gmail.com).

http://dx.doi.org/10.17562/PB-53-8

ISSN 2395-8618

in the main index, if it is indexed. To discover additional information like documents from people who are not users of the system, or documents where these people play an important role, they can be modeled following ontologies.

To make modifications that allow us to find new relations among the objects that are part of the system is not a simple task if it is made from the DB. For this reason, the use of ontologies is proposed to create models that define new relations and provide more information in the system.

In an ECM, documents are handled as such. Documents are understood as any form to present information, no matter its format or content. The ECM can manage resumes, contracts, invoices, mails, PQR, brochures, recipes, reports, researches, etc. It does not matter either its digital format (word, Excel, PDF).

Considering this variety, the ECMs choose to create metadata common to all, modeling them as a document. For this, metadata are created following some schemes that allow them to organize in hierarchy the information and save related information with characteristics of the document (physical location, format, entry date in the system).

In a search engine based on key words as ECMs commonly do, generally there are failures when alias or pseudonyms and names changes are handled. Another problem arises when dates not handled by the metadata are searched and they are in different formats. For instance, we will not get the same results from the date “01/23/2013” as we look for “Twenty Third Day of January 2013”, even if they make reference to the same date.

With the creation of an ontological model, basic relations of hierarchy can be established, as well as more elaborated relations that allow to associate objects of different classes [7].

III. Related works

The semantic web is a group of techniques and technologies to represent the knowledge in a specific domain [8]. These techniques allow sharing and reusing the information among applications and communities. Technologies such as RDF (Resource Description Framework)[9] and OWL (Ontology Web Language) to represent knowledge [10] are used by companies to create software like Athento [1] and Zaizi [2], which add an improvement to the search engine of the ECM system.

The traditional search engines base their operation in the use of inverted indexes, which enable a high speed of response [11]. With the use of semantic, and indexing documents with relevant search terms (relevant to each document), allow Athento users to navigate through documents that are related.

In the works currently under development, it is always attempted to apply or develop an ontological model to represent the domain managed by the ECM like in the OpenCalais project [12]. In this work, it is pretended to handle the ambiguities and alias that could be present in the text of the document.

Some authors propose the use of ontology in ECM software to manage the problem of ambiguous representation of knowledge with two approaches, ex post and ex ante. Ex post try to solve the ambiguity once the information has been collected, on the other hand ex ante try to avoid the ambiguity before it happens [13].

The use of ontology models can help to build a structure that represents the possible class that a document could belong to, and can be used to classified documents in a repository. Instances of the ontology model can be use to tag the content too, and could be used to let the final user choose the document type that he or she like the most [14].

Some collaborative work require the interchange of recorded data to accomplished a job, most of the time is hard to share with collaborators or to reuse the data (or information) in some other process, because the data is not in the correct format or can has different meaning from one data base to other. Some authors propose to solve this problem by using an information system in conjunction with ontology models to give an easy way to access the information [15]. In this approach the use of ontology can help to build a semantic tag system to accurately annotate the document from repository, and give the final user the ability to access the knowledge present on the ontological relations [15]. The main objective of this approach is to let the final user spend less time on preprocessing the data for exchange with coworker, or reuse the data in some other different process.

Semantic and ontology models are used to give a web page a more structure and search engine friendly form. The W3C has proposed some tags that help the web programmer to build more structured web page. For example, it is possible to use the tags <article> reference, article, and comments about that article. By using this tag is possible to share and reuse data with applications, enterprises, and communities [16].

It is possible to treat the ambiguity with the creation of classes in the ontological model and declaring them as disjointed. For example, with the word blackberry, which can make reference to a cell phone brand or a fruit, the classes shown in Figure 1 could be created.

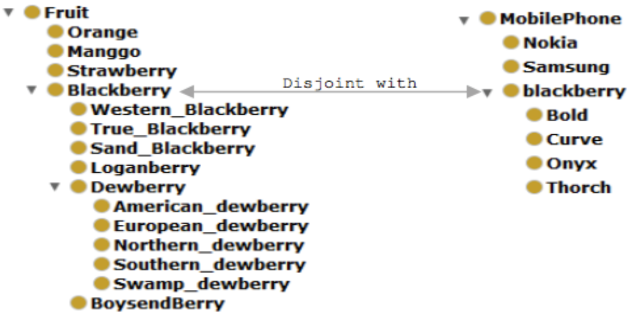


Fig. 1. Disjoint Class example.

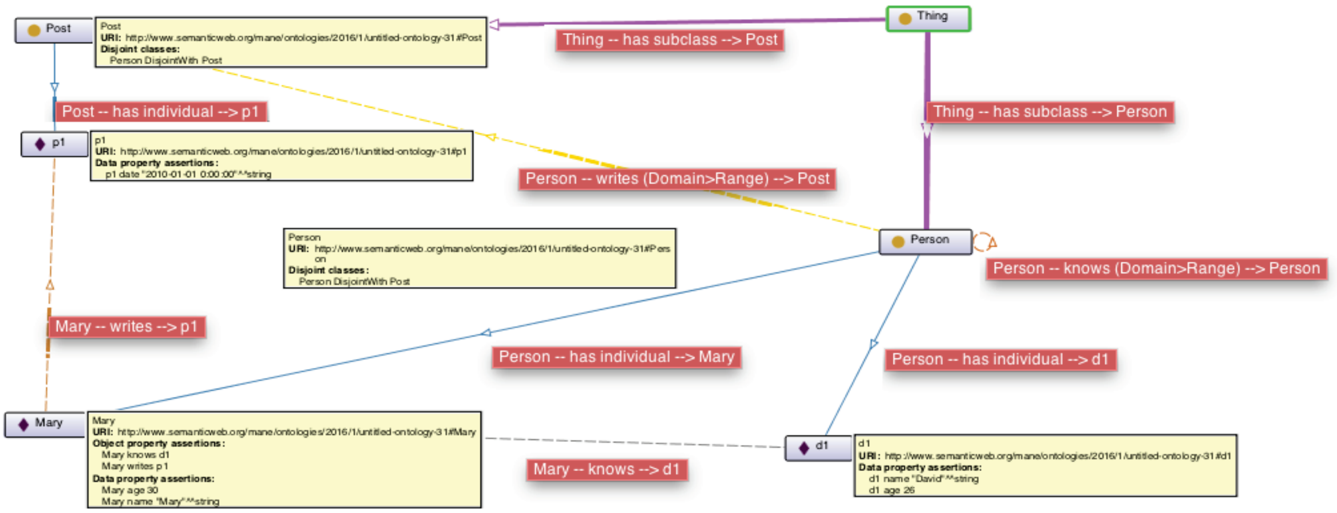

Post
Post
URI: http://www.semanticweb.org/mane/ontologies/2016/1/untitled-ontology-31#Post
Disjoint classes:
Person DisjointWith Post
Thing
Thing -- has subclass --> Post
Thing -- has subclass --> Person
Post -- has individual --> p1
p1
p1
URI: http://www.semanticweb.org/mane/ontologies/2016/1/untitled-ontology-31#p1
Data property assertions:
p1 date "2010-01-01 0:00:00"^^string
Person -- writes (Domain>Range) --> Post
Person
URI: http://www.semanticweb.org/mane/ontologies/2016/1/untitled-ontology-31#Person
Disjoint classes:
Person DisjointWith Post
Person
Person -- knows (Domain>Range) --> Person
Mary -- writes --> p1
Person -- has individual --> Mary
Person -- has individual --> d1
Mary
Mary
URI: http://www.semanticweb.org/mane/ontologies/2016/1/untitled-ontology-31#Mary
Object property assertions:
Mary knows d1
Mary writes p1
Data property assertions:
Mary age 30
Mary name "Mary"^^string
Mary -- knows --> d1
d1
d1
URI: http://www.semanticweb.org/mane/ontologies/2016/1/untitled-ontology-31#d1
Data property assertions:
d1 name "David"^^string
d1 age 26

Fig. 2. Example of ontology model with Protégé

IV. Ontology

Ontologies and semantics have become a very important subject in the last years, which are researched by different academic groups. Ontologies are used in software design to establish communication among actors of design, interfaces and communications design, and knowledge discovering [17].

To create an ontology model we can follow the same steps as Object Oriented software design, to create a formal specification of the terms that belongs to a domain. To represent all this information we can use classes, attributes, relations, and instances. The relations let us express how objects from the domain are related with objects of the range.

We can build hierarchy with classes and sub classes, and explicit express what classes are disjoint, for example on figure 2 we have Person class and Post class, that are disjoint and are related by the relations “writes”; this relation express that a person writes a post. The dotted lines express relation with objects of the same class, for example “knows” shows the relation between persons.

The studies for the use of ontologies are made in order to apply them in a legal environment [18], considering as a base the definitions (content, intellectual property, instantiation) provided by the Dublin Core framework [19], in order to have information of the information, but the idea is not only focused on using the hierarchy developed with the model, but also to create relations that can provide more knowledge [18].

Ontologies are also used to share knowledge among systems, to allow the communication among intelligent agents, and in the software development to identify requirements and set tasks [20].

V. Proposed Approach

The use of ontological models is proposed to manage the problems previously mentioned. For this purpose, it is necessary to create instances of the ontological models with the information of the entities presented in the text of each document. In order to control or handle the alias, the use of the relation “theSameAs” and the relation “hasFacet” to handle the ambiguities is proposed. These relations must be integrated in the ontological model and used at the moment of the creation of instances. Once instances are created, they are indexed in order to be found quickly at the moment of the search. All this is based on the communication between the ECM (using the communication CMIS standard of the ECMs) and a module to create ontological instances.

In an ECM, the nature of documents can be varied and depends on the use given by the company that uses them. We can find recipes, invoices, resumes and articles. The use of a unique model would not be good, because it could leave out entities that represent important information. As we can observe in Figure 3, a model to represent people, institutions and publications with the relation “theSameAs”, is used.

A good basis for the ontological models is FOAF [3], which represents the relations of people (friendOf, fatherOf, etc.) and their basic data (name, last name, etc.), then the system can be enriched with models such as resumeRDF [21] that represents the information of the resume and organization [8], which gathers the information about the organizations that could be related to a person.

A. Alias handle

The problem of the alias arises when a document inside of the ECM uses an alias to make reference to an entity in the system. The alias is identified as an instance in the model, and a relation “theSameAs” is created between the alias and the instance referred to.

To make the search the following steps must be followed:

ISSN 2395-8618

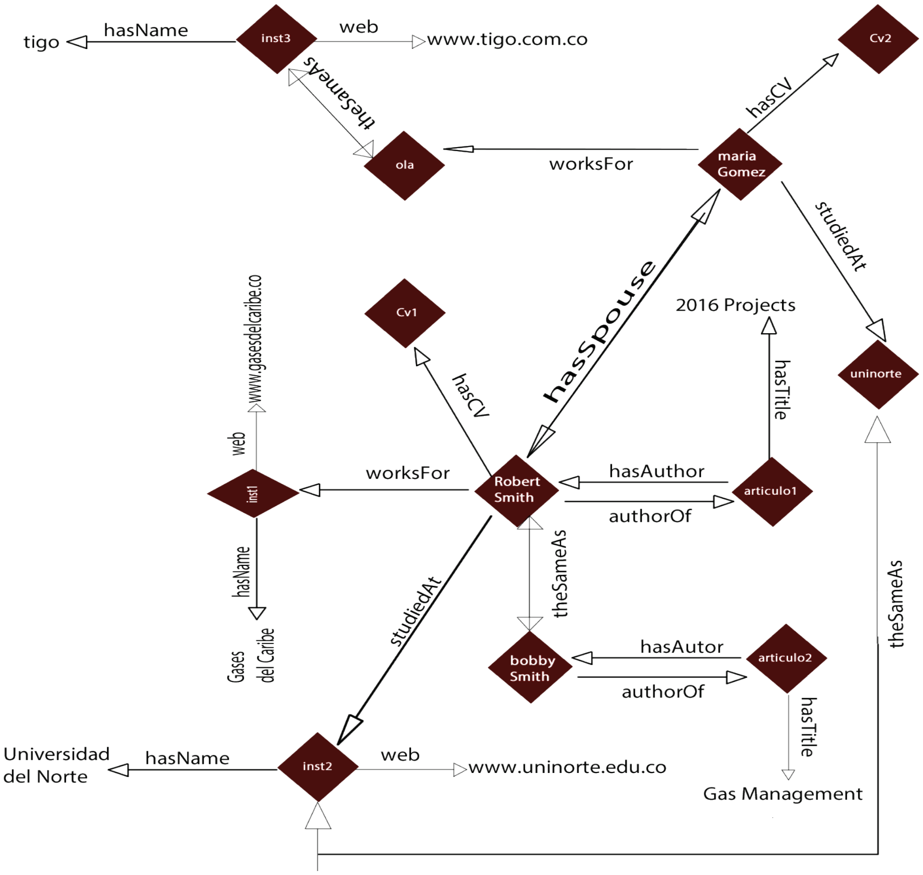


Fig. 3. Instance of ontological model with alias.

1. Indexes (where the texts of each document are indexed) are consulted, and a list of results with the found coincidences is created.
2. Indexes of the ontological models instances are checked:

   2.1 Instances whose name matches with the searched words are consulted, and their relations with other individuals are added to a list.

   2.2 The individual obtained from the previous step is consulted using "theSameAs". Then the relations and properties of most interest (authorOf) are consulted and added to the list.
3. The obtained results are organized in two lists and presented to the user.

In Figure 3, "RobertSmith" and "BobbySmith" entities make reference to the same person. If the user makes a search with the words "Robert Smith", the relation "theSameAs" between the two entities allows recommending the document "Gas management".

B. *Handling ambiguity*

Ambiguity arises when the user searches a word that can have more than one meaning, and the search engine shows as result any document that contains the searched word, without taking in consideration the meaning of the word in the text [22]. With the relation "hasFacet" in an ontological model, this situation can be handled and represent the different meanings that a word can have.

In order to show to the final user the different meanings that a word can have, these steps must be followed:

1. Indexes (where the text of each document is indexed) are consulted, and a list of results with the found coincidences is created.
2. Indexes of the ontological models instances are checked:

   2.1 Instances whose name matches with the searched words are consulted, and their relations with other individuals are added to a list.

   2.2 The individual obtained from the previous step is consulted using "hasFacet". Then the relations

ISSN 2395-8618

and properties of most interest (authorOf) are consulted and added to the list.

3. The obtained results are organized in two lists and presented to the user.

The goal is to have a recommendation system based on ontologies for an ECM, but these steps here described could be used in any search engine after making the necessary changes.

In the tests, Abox ECM [23] has been used, a web application ran under Win7, Sqlserver [24] and .Net framework 4.5 [24]. All the code for the handling of ontologies and the instances was developed with C# [25] and the library dotNetRDF [26]. The recommendations system was developed following the design pattern MVC in order to be shown inside of the application Abox [23]. A controller that makes the process previously described was developed, which communicates with the ECM by means of the CMIS standard (Figure 4). A view was also created, and whose principal task is to create a <div> block (Figure 5) with the recommendations. The work and handling of the ontological models were developed with the tool Protégé [27, 28].

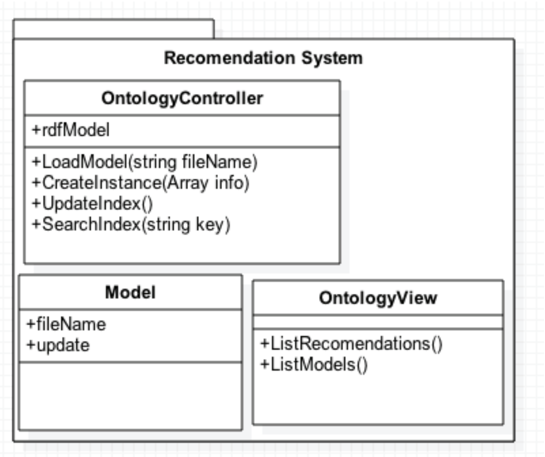
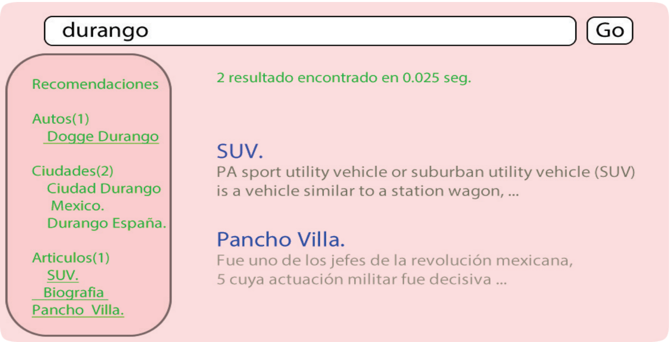


Fig. 4. Diagram of the system.



Recomendaciones

Autos(1)
Dogge Durango

Ciudades(2)
Ciudad Durango
Mexico.
Durango España.

Articulos(1)
SUV.
Biografia
Pancho Villa.

2 resultado encontrado en 0.025 seg.

SUV.
PA sport utility vehicle or suburban utility vehicle (SUV) is a vehicle similar to a station wagon, ...

Pancho Villa.
Fue uno de los jefes de la revolución mexicana,
5 cuya actuación militar fue decisiva ...

Fig. 5. Result view with <DIV> block(on the left) that have the recommendations.

VI. Results

All the tests were conducted with the ECM Abox [23], a repository of 4322 documents, the FOAF models [3], organization, and place [29].

In the repository, there are some documents with ambiguities, and non-related documents, but in the text, there is the word "Durango", which makes reference to a place in Mexico, a place in Spain, and a car model.

To handle this case, the "Durango" entity was created, and the relation "hasFacet" was used for each of the different instances referred to.

Using the instances of the Figure 6, it is possible to suggest the user all the possible meanings of the word "Durango" inside of the system.

In Figure 5, it is presented the HTML view of the Abox searcher in which the list of recommendations is embedded in. It can be observed that the results obtained with the traditional system (right) are not clear for the user. The system has found a total of 15 documents that have to be read by the user to discard those that are not related to the search. In the left side, the list of recommendations created by the proposed system can be observed and which presents to the user the different aspects registered in the system regarding the searched word. With this list, the user can discard as quickly as possible (they do not have to read the unnecessary documents) the documents not related to its search.

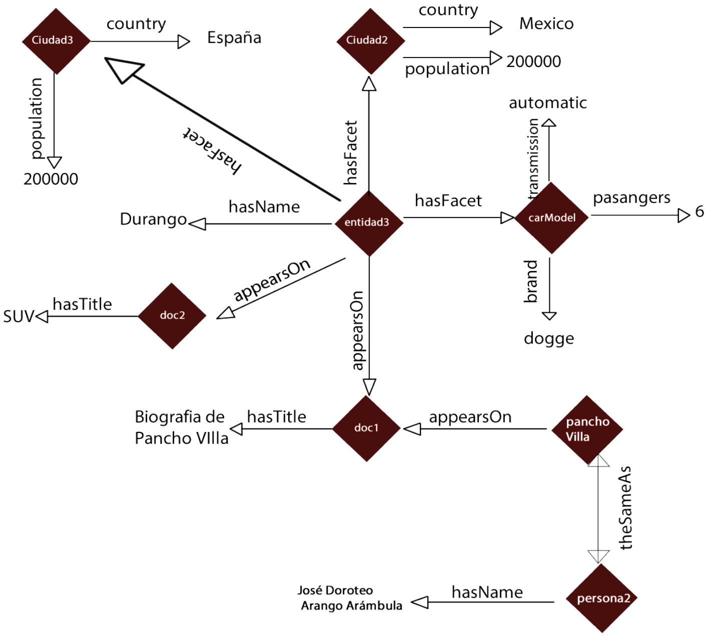


Fig. 6. Instance of ontological model with ambiguity.

VII. Conclusions

Showing the relation among entities that appear in the text of a document is an advantage for the ECM user.

In this development, model FOAF [3] is used because it has many relations, but the steps here described can be used with any ontological model, and it is even recommended to use multiple models with the purpose of representing the largest amount of entities that have a relevant meaning in the business logic of the system.

Providing suggestions to the user helps him to make decisions which documents are relevant for its search and save time. In this work, ambiguities are handled with a relation in the ontological model, but it is also possible to do it by defining a class and a subclass for every meaning that a word can have, which is not practical because it should previously be known which are the ambiguous terms and their different meanings.

We show in this publication the advantages of the use of ontologies, not only to represent metadata, but also to represent the entities present in the text of the document.

Acknowledgments

This work was supported by the Departamento Administrativo de Ciencia, Tecnología e Innovación (COLCIENCIAS, Colombia) through 562-2012 Banco de Proyectos y Programas I+D+I – Co-funding Modality, which funds the Project 121556236053.

References

[1] "Software ECM | Athento." [Online]. Available: http://www.athento.com/software-enterprise-content-management/. [Accessed: 07-Jul-2015].

[2] "Home | Zaizi." [Online]. Available: http://www.zaizi.com/. [Accessed: 07-Jul-2015].

[3] "The FOAF Project." [Online]. Available: http://www.foaf-project.org/. [Accessed: 07-Jul-2015].

[4] "Car sales Ontology" [Online]. Avilable: http://www.heppnetz.de/ ontologies/vso/ns . [Accesed: 15- Jan - 2015]

[5] "GeonNames Ontology" [Online], Avilable: http://www.geonames.org/ ontology/documentation.html . [Accessed: 10-Jan-2015]

[6] "AIIM - The Global Community of Information Professionals." [Online]. Available: http://www.aiim.org/. [Accessed: 29-Apr-2015].

[7] G. Barchini, M. Álvarez, and S. Herrera, "Information systems: new ontology-based scenarios," *JISTEM-J. Inf. Syst. Technol. Manag.*, vol. 3, no. 1, pp. 2–18, 2006.

[8] "World Wide Web Consortium (W3C)." [Online]. Available: http://www.w3.org/. [Accessed: 07-Jul-2015].

[9] "RDF - Semantic Web Standards." [Online]. Available: http://www.w3.org/RDF/. [Accessed: 29-Apr-2015].

[10] G. Antoniou and F. Van Harmelen, *A semantic web primer*. MIT press, 2004.

[11] S. Brin and L. Page, "Reprint of: The anatomy of a large-scale hypertextual web search engine," *Comput. Netw.*, vol. 56, no. 18, pp. 3825–3833, 2012.

[12] "Thomson Reuters | Open Calais." [Online]. Available: http://new.opencalais.com/. [Accessed: 07-Jul-2015].

[13] Jan Von Brocke, "Enterprise content management in information system research, foundations, methods and cases", Springer, 2014.

[14] Daniela Briola et al. 2013, "Ontologies in industrial Enterprise Content Management Systems: the EC2M Project".

[15] Abdelkader Hameurlain et al. "Transaction on Large-Scale Data- and knowledge-centered systems IV". Springer, 2011.

[16] W3C semantic elements. [Online]. Available: http://www.w3schools.com/html/html5_semantic_elements.asp . [Accessed: 9-Jul-2015].

[17] G. N. Aranda and F. Ruiz, "Clasificación y ejemplos del uso de ontologías en Ingeniería del Software," in *XI Congreso Argentino de Ciencias de la Computación*, 2005.

[18] D. Tiscornia, "The LOIS project: Lexical ontologies for legal information sharing," in *Proceedings of the V Legislative XML Workshop*, 2006, pp. 189–204.

[19] "DCMI Home: Dublin Core® Metadata Initiative (DCMI)." [Online]. Available: http://dublincore.org/. [Accessed: 07-Jul-2015].

[20] T. R. Gruber, "A translation approach to portable ontology specifications," *Knowl. Acquis.*, vol. 5, no. 2, pp. 199–220, 1993.

[21] "ResumeRDF Ontology Specification." [Online]. Available: http://rdfs.org/resume-rdf/. [Accessed: 07-Jul-2015].

[22] J. Lyons, *Linguistic semantics: An introduction*. Cambridge University Press, 1995.

[23] "ECM Abox | Adapting." [Online]. Available: http://www.adapting.com/index.php/abox-ecm/. [Accessed: 07-Jul-2015].

[24] "Microsoft – Official Home Page." [Online]. Available: https://www.microsoft.com/en-gulf/. [Accessed: 07-Jul-2015].

[25] "Visual C#." [Online]. Available: https://msdn.microsoft.com/en-us/library/kx37x362.aspx. [Accessed: 29-Apr-2015].

[26] "dotNetRDF - Semantic Web, RDF and SPARQL Library for C#/.Net." [Online]. Available: http://www.dotnetrdf.org/. [Accessed: 07-Jul-2015].

[27] N. F. Noy and D. L. McGuinness, "Desarrollo de Ontologías-101: Guía para crear tu primera ontología," 2005.

[28] "Protégé." [Online]. Available: http://protege.stanford.edu/. [Accessed: 29-Apr-2015].

[29] "Organization - schema.org." [Online]. Available: https://schema.org/ Organization. [Accessed: 07-Jul-2015].