Ontologies in industrial Enterprise Content Management Systems: the EC^{2}M Project

Daniela Briola
DIBRIS
Genoa University
Genoa, Italy
daniela.briola@unige.it

Alessandro Amicone
Nacon Information Technology
Sempla Group
Genoa, Italy
alessandro.amicone@nacon.it

Dante Laudisa
SEMPLA
Sempla Group
Milan, Italy
dante.laudisa@sempla.it

***Abstract*—Enterprise Content Management (ECM) systems represent a crucial aspect in the efficient and effective management of large-scale enterprises, in particular for those made up of several sites distributed all over the world. Considering the increasing number of documents that large enterprises need to store and manage, and the need of classifying and retrieving them in real time, ECM systems need to be strengthened to represent not only standard "syntactic" information associated with documents but also more complex and structured information to represent the documents' semantic. In this paper, we present a joint project of the Department of Informatics, Bioengineering, Robotics and System Engineering of the University of Genoa, Italy, and two companies, Nacon (part of Sempla Group) and Nis, to create an improved ECM system (named EC^{2}M) exploiting ontologies to better classify, retrieve and share documentation among different sites of the involved companies. The overall architecture of the EC^{2}M system is presented with a detailed description of the ontology that has been created for the first EC^{2}M prototype, modeling the Sempla's business offers, to let users classify their documents in a semantically-driven way and search them using semantic tags. EC^{2}M is "parametric" in the used ontology: the system is currently used by Sempla, with the ontology presented in this paper.**

Keywords- Enterprise Content Management; Ontologies; Semantic Classification; Knowledge Representation; Industrial Application

I. Introduction and state of the art

The international Association for Information and Image Management (AIIM), the worldwide association for enterprise content management, defined the term "Enterprise Content Management" in 2000, but it has been updated many times to adhere to the continuous new market needs. The more recent definition is: *Enterprise Content Management (ECM) is the strategies, methods and tools used to capture, manage, store, preserve, and deliver content and documents related to organizational processes. ECM covers the management of information within the entire scope of an enterprise whether that information is in the form of a paper document, an electronic file, a database print stream, or even an email* [1]. ECM is an "umbrella term" covering document management, web content management, information search, collaboration, records management and many other tasks, but it is primarily aimed at managing the life-cycle of information, from initial publication or creation to its disposal, to preserve a company's internal (often unstructured) information, in all of its forms. Therefore, most ECM solutions focus on Business-to-Employee (B2E) systems, but nowadays, thanks to the improvement of the IT capabilities and because of the increasing users' need to classify documents according to their meaning, these systems have grown in complexity and often integrate modules to exploit more structured information, taxonomies, dictionaries and so on. Many vendors are offering products in this area, starting from the commercial ones (Microsoft, IBM and Oracle) moving to many powerful open source solutions (for example Alfresco [2], Plone [3] and SenseNet [4]). The new trend in this field is to create ECM systems that can automatically extract information from documents to classify them or add a semantic layer to tag documents in a more structured and interesting way: this is the area where the EC^{2}M project is located.

The problem of classifying, retrieving and sharing documents among users and companies pertains to the research field of knowledge sharing, whereas the problem of semantically tagging documents pertains to that of the knowledge representation. Both fields are relevant both from an academic and an industrial viewpoint, and this motivates the joint academic-industry EC^{2}M project. In EC^{2}M, we used ontologies as a way to structure information describing documents and their content. Many similar studies and projects have been conducted in this area: an example of a commercial ECM system semantically enriched is SmartLogic [5], that offers an automatic classification, based on an automatically-extracted taxonomy of documents. Many open source systems have been developed to integrate semantic services in the document/information management, among which H-DOSE ([6]), OPEN-CALAIS ([7]) and APACHE STANBOL ([8]). Even if they are not ECM systems according to the standard definition, they deal with very similar problems.

If we look at academic research, we can cite for example [9] (describing the Rhizomer CMS, that tags its items using semantic metadata semi-automatically extracted from multimedia sources), [10] (that proposes a framework to manage and share written information contents using an ad-hoc knowledge model for an industrial research center), and

[11] (that presents an open architecture framework based on the open-source CMS OpenCMS and a Java-based web management system for learning objects, which were derived from the instructional materials used in several postgraduate courses).

Our system exploits some of the above-mentioned open-source systems and integrates an ad hoc ontology, shared among the different nodes of the network, to model the documents types and their content. The system offers a publish/subscribe service and is based on a cloud platform. It also exploits contextual information on the location and device used by the user to implement context-awareness. EC^2M is thus a concrete industrial example of how these technologies can be coordinated to create a new powerful system, that can be actually used by existing enterprises.

The rest of the paper is organized in four sections: Section II describes the the EC^2M system, Section III presents the ontology, Section IV shows the prototype and, finally, in Section V, some considerations and conclusions are presented.

II. The EC^2M System

The Enterprise Cloud Content Management (EC^2M) system was born from the collaboration among the Department of Informatics, Bioengineering, Robotics and System Engineering of the University of Genoa and two outstanding IT Italian enterprises, namely Nacon (member of the Sempla Group and specialized in the design and implementation of complex systems, ECMs, and process management) and Network Integration & Solutions (Nis), specialized in the design and development of network products and services for businesses, public administration and end-users.

The developed system is a Content Management System that aims to semi-automatically classify documents with respect to a set of predefined tags: these documents will be then shared among different partners (called "Nodes of the EC^2M network", that are companies, or companies' sites, that need to collaborate and chose to agree on the common ontology to tag documents), located in various physical locations, in an automatic way.

The types of documents and their possible contents are modeled using an ontology that formally describes them; the ontology instances are used to tag the documents with semantic information. Every user in the system is able to subscribe to a set of "interests" (chosen from the instances in the ontology) so that when a new document is inserted in the EC^2M network and tagged (manually or partially automatically) with terms from the ontology, those users that are interested in those terms are pro-actively informed that a new document is available. A centralized "semantic router" is in charge of sharing these documents and the system is implemented exploiting a Cloud platform.

A software module manages the context (location and device) where the user is acting, to give the user a subset of the information he needs considering the device he is working on.

The system is deployed over the Cloud Amazon Web Services (AWS) platform (using it as an "Infrastructure as a Service"), that is a good compromise between cost and performances. This solution allows to simply scale the number of nodes in the EC^2M network or to scale the physical resources used to manage the network, to get better performances. The EC^2M system has been designed to run on a private physical network too.

The EC^2M system offers "internal services" to individual nodes (corresponding to an enterprise site) and "external services" to let nodes interact. Looking at individual nodes, the so called "semantic publish/subscribe" service allows every user to declare the arguments he is interested in, chosen from those described in the ontology, and then makes available (globally on the network and locally to the node) the documents matching the subscribed interests.

Looking at the complete network, that is, at the services connecting different nodes, the aim of the system is to allow users from different nodes to be informed of documents, on other nodes, that are interesting for them. This is where the ontology comes into play: sharing interesting documents across nodes is in fact possible because the nodes share a common ontology.

Every node in the network may have privacy policies, because not every document of a node should be read by all the other nodes: maybe only some information as title, abstracts, etc can be shared. These policies are managed by the nodes. Issues related to policy management are out of scope of this paper and are not described here.

Every document is characterized by a set of standard attributes (or tags), like its Name, Creation date, Abstract and so on, whereas the document type is chosen from the ontology: then the user can add other tags in a manual or semi-automatic way (see more details in Section IV), selecting the values from the instances of the ontology and driven by their relationships.

The EC^2M system can be "instantiated" many times, to be useful for different groups of enterprises (that is, for new enterprises' networks): a new ontology, describing types of documents and their possible contents must be created, but the overall structure and behavior remain the same. In this sense, EC^2M is "parametric" in the used ontology.

At a high level of abstraction, the system is divided into different modules (see Figure 1):

- Graphical User Interface (GUI): front end of the system, where users can log in and insert/retrieve documents (based on the ontology);
- Crawler: software that explores predefined hard disk sectors to find documents that are sent to the Loader;
- Loader (or Classifier): module that automatically extracts from the documents tags corresponding to those in the ontology and that enriches the documents with

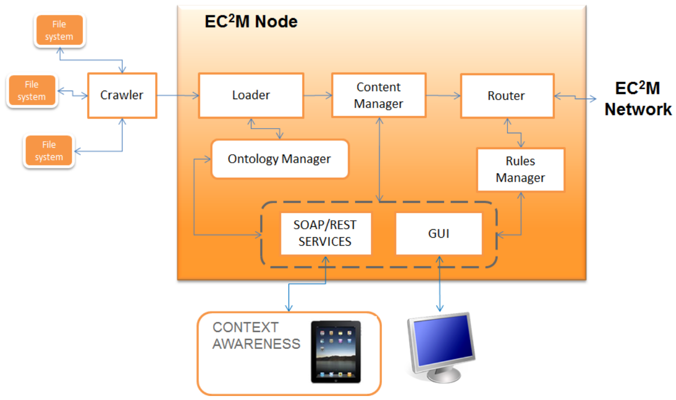

Figure 1. The high level architecture of the EC^{2}M system.

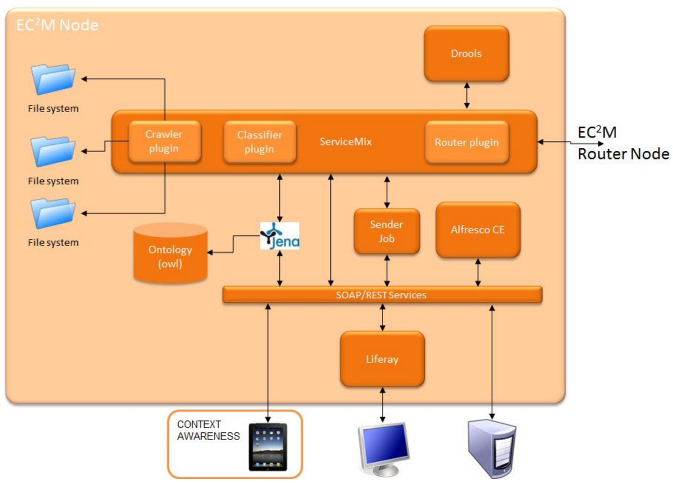

Figure 2. The software architecture of the EC^{2}M prototype.

tags related to those already manually associated with the document, using the instances in the ontology and their relationships;

- Ontology Manager: module that queries and manages the ontology;
- Content Manager: module that stores the documents and manages their sharing and retrieval;
- Router: module that manages the sharing of documents between nodes;
- Rules manager: module that the Router uses in order to define and dynamically create routing rules.

In Figure 2, the reader can see which (existing or new) software modules have been used to implement the first prototype of the EC^{2}M system. On the right side, the arrow points to the EC^{2}M Networks that is not reported in the figure and that is the core of the routing system. Since it is not in the scope of this paper, as the other modules not strictly connected to the ontology, it has been left out from the figure for readability.

To physically manage the documents, we adopted Alfresco, that is a well known open source Java Content Management System: this system allowed us to exploit all the facilities of a high performances business platform with the good property of being an open source software. Furthermore, Alfresco takes advantage of many other well known open source systems like Spring, Hibernate, Lucene and MyFaces.

To create and manage the ontology we adopted the Web Ontology Language (OWL [12]) and Protégé [13]. The ontology is not subject to frequent changes: if it needs to be modified, this is done using Protégé and then the new version is again made available to the framework for queries.

The other modules of the node are not described in details because out of scope, but they are shown anyway in Figure 2 to offer a complete view of the node implementation.

III. THE ONTOLOGY

The EC^{2}M system was designed to work with any ontology describing the documents to be shared and their contents. As a concrete example, we decided to design, implement and use an ontology modeling the Sempla's business proposal.

Sempla, as a brand, was founded in 2009 and operates in System Integration and Information Technology consulting. It has nearly three decades of experience with the most important Italian Groups from the Financial, Production and Public sectors.

This domain was chosen because Sempla is a very large enterprise, covering different business areas, so its documentation presents many types of documents and a large set of terms that are of interest for different users. These terms and types of documents are quite common in this business area, so modeling the Sempla domain is the best choice because the emerging ontology is correct also for Nacon (that is member of the Sempla Group) and for Nis (that often collaborates with Nacon so can easily adhere to the ontology), which are the other nodes in the first prototype.

A. The Ontology Design

To model the domain with an ontology (as defined in [14]), we adopted the Noy and McGuinness methodology [15], that being an agile method is very suitable for collaborating with industrial partners. This methodology foresees these stages:

1) determine the domain and scope of the ontology;
2) consider reusing existing ontologies;
3) enumerate important terms in the ontology;
4) define the classes and the class hierarchy;
5) define the properties of classes-slots;
6) define the facets of the slots;
7) create instances.

The first step was quite simple to follow: the domain of the ontology was the Sempla's business proposal. It is translated into a complex organization of the logic concepts

 ISBN: 978-1-61208-273-8

that describe what Sempla offers to its costumers, in terms of products and high technical and management consultancy.

Documents must be tagged to describe, with instances found in the ontology, their structure (some type of documents can have many attachments) and above all their technical and business items: for every business market Sempla has a “portofolio” of products and consultancy services that is well organized and defined.

We searched for similar ontologies, but we were not able to find one that was useful for modeling our domain. Maybe other companies own a similar ontology, but they are not public. We also considered existing ontologies, for example Bibo ontology [16], but we did not use them because they share only very few terms with those used in our domain. We could use the already exiting Sempla’s documentation, that offered us an already well-defined set of terms describing the domain, although in natural language and not structured in any standard format.

The third step was conducted with the collaboration of the domain experts, that were the scientists from Sempla, Nacon and Nis. The majority of the terms was collected analyzing the brochures describing Sempla’s business proposal (one is summarized in Table I) as well as a large set of documents selected as example from the real ones and a list of terms created by the “users-to-be” of the system, that listed the terms (corresponding to logic concepts used in their work) that they would like to use to classify a document.

To define the classes and their structure, we asked the domain experts to describe in details the types of documents they use, the most relevant information characterizing them and how they model the different business markets. We also took inspiration from the file system where documents were stored: the directories were partially organized as the business areas, and this organization reflected the way Sempla divides its business proposal.

The definition of the properties was done following a similar process.

As a last step, we manually inserted the instances in the ontology: the instances are named using some of the terms listed before, and the properties join them to completely describe the domain. It should be noted that in this process, stating what had to become a class and what an instance was a complex task, because some domain’s logical concepts may be mapped into a class (if we foresee a possible future extension) or into an instance as well (because they are something already stable and with different properties values), for example for *Market*’s instances. In these cases, we must consider that only instances will be used to tag documents, so it was an obliged decision to model those terms as instances.

Since the ontology is aimed at being used by Italian users it was created in Italian, but some terms (in particular the instances’ names) are in English, to maintain a link with the existing documentation and Sempla’s internal standards.

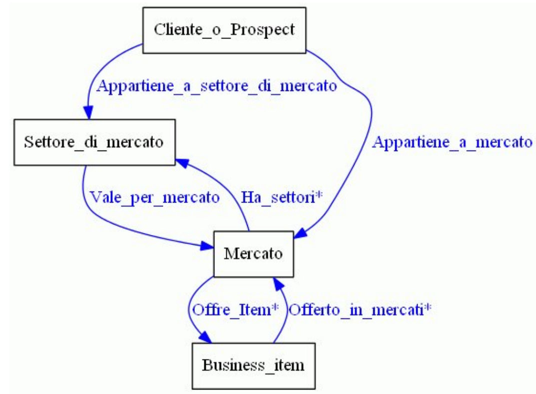

Figure 4. Class *Mercato* and its main relationships.

B. The ontology details

The Sempla business proposal is organized considering different business markets, to propose ad hoc solutions for each area. Each high level business market is called *Mercato* (Market), and each *Mercato* is characterized by some distinct *Settore* (Sectors). To give and explanation of what we assume to be a market, consider that its instances are “Product”, “Financial Services” and “Insurance”, describing the main activities of the costumers operating in each market. Starting from this macro division of areas, all the other classes are related and organized considering these three sectors. For example, each costumer (class *Cliente_o_Prospect*) refers to one sector, so that his market is uniquely identified.

The Sempla business proposal is created combining different items, identified with the class *Business_Item*, that are divided into different types (subclasses) and that refer to six different Business Areas (as shown in Table I), modeled with the class *Area_Business*, with instances: “Business_IT_consulting”, “Business_project_outsourcing”, “Web_digital_design”, “Business_Solution”, “IT_services”, “IT_solution”. Every *Business_Item* can be associated with only one *Area_business* but can be offered to different Markets. A business item that is offered in many markets is called “cross market”. The class that groups these items is defined with a necessary and sufficient condition, and is called *BI_Cross_Mercato*. In a similar way, the class *AB_cross_mercato* is defined with the necessary and sufficient condition that it collects the business areas offering at least one “cross market” item.

The business items are divided into four disjoint subclasses (*Prodotto*, *Soluzione*, *Servizio*, *Attivitá_professionale*), and are related with class *Ambito_Tecnico* describing at high level the IT area they refer to.

In Figures 3 and 4, the relationships between the classes described above are shown.

The documents that must be classified are divided into different types, modeled with the classes (subclasses of

Table I
THE SEMPLA BUSINESS PROPOSAL FOR THE "FINANCIAL SERVICES" MARKET, GROUPED BY BUSINESS AREAS.

Business IT Consulting	**Digital Marketing & Design**	**Business Solution**	**IT Solutions**	**IT Services**	**BPO**
BPR; Studi di fattibilitá; Enterprise architecture planning; Program management consulting; Organizzazione processi IT; Project portfolio management.	Web & Content Design; User Experience; Community management; Digital Advertising; Augmented Experience.	Credit & Risk Management; Filiale a CRM; Contact center; Pagamenti, Monetica, ATM/POS; Finance & Wealth Planning; Controlli e compliance; Sicurezza e Antifrode; Tesoreria; Human Resources; Reporting & Business Intelligence; Credito al consumo; Leasing & Factoring; Banca Virtuale; Project Portfolio; General ledger.	System Integration Framework; Application Frameworks; Metodologie di Delivery; Multicanalitá; Enterprise Content Management; DB Administrator.	Application Management; Application Modernization; IT Infrastructure Management; ITIL Implementation.	Contact Centre; Back Office; Fiscalitá Locale; Postalizzazione; Business travel management; Formazione, RollOut, Help Desk.

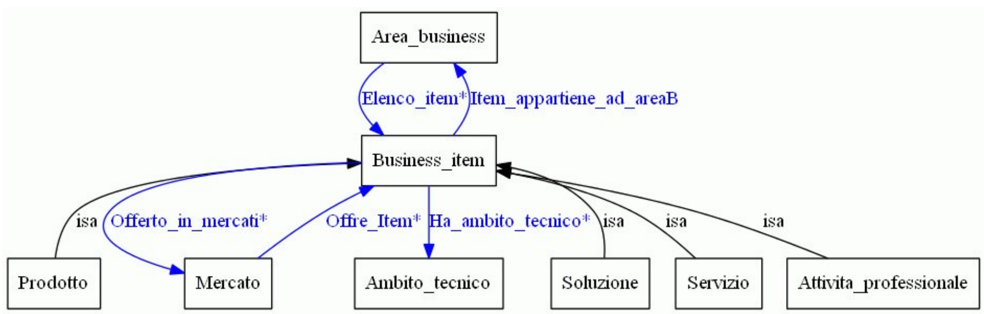

Figure 3. Class *Business_Item* and its main relationships.

Documento (Document)):

- Proposal (class *Documento_offerta*): it describes a business proposal to a costumer;
- Attachment (class *Allegato*) with its sublcasses *Allegato_generico* (generic attachment) and *Allegato_tecnico_offerta* (technical attachment of one specific proposal);
- Consultation document (class *Documento_di_consultazione*), representing documents available for information;
- Request for proposal (class *Request_for_proposal*) coming from a costumer (generally these documents are related to one or more Proposals);
- Marketing support documentation (*Supporto_marketing*), that is a document that describes a costumer and its business;
- Presentation (class *Presentazione*) divided into:
  - Presentation pertinent to a proposal (class *Presentazione_allegata_offerta*), that is a presentation prepared for a specific Proposal, and that can be related to some *Business_item*; and
  - Society presentation (class *Presentazione_istituzionale*) that describes the Sempla business proposal to a specific Market (that must be specified).

In Figure 5, an overview of the relationships among the different types of documents is reported, while in Figure 6 a detailed view of the relationships among *Proposal* and the other classes is shown.

We also created the classes *Ambito_Funzionale* and *Partnes*, representing a further classification of the business services from a commercial viewpoint and a list of possible third parties companies involved in the proposal, respectively.

Due to space limitation, we only show properties in figures and do not report instances of every class. In Section IV the reader can find some of these instances shown in the figures, while they are used by the EC^{2}M interface to help user tagging a document.

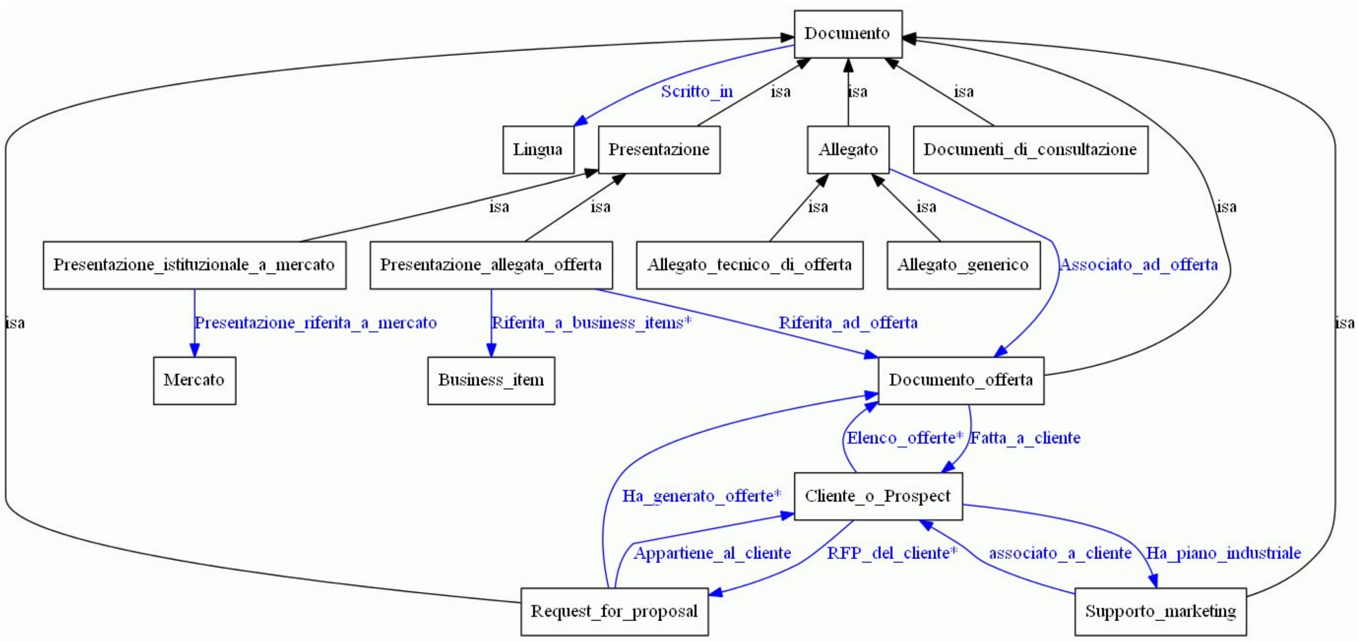

Figure 5. Relationships among classes starting from *Documento* (*Document*) and its subclasses.

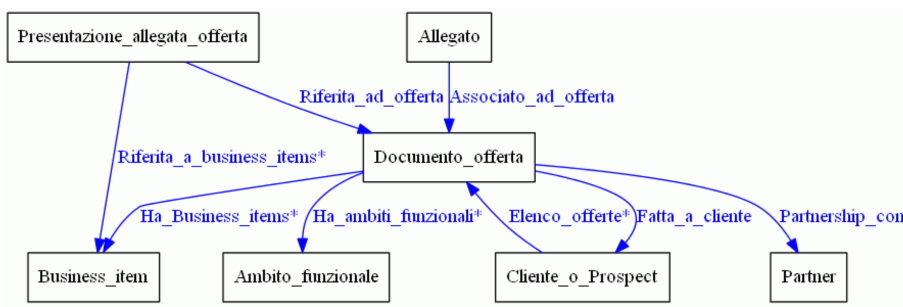

Figure 6. Class *Documento_offerta* (*Proposal*) and its main relationships.

IV. THE PROTOTYPE

A first prototype of the EC^{2}M system has been developed at the end of 2012, based on the ontology described in Section III, and has been tested concretely by Nacon, Nis and Sempla to manage the documents created in 2012 (3200 documents), with 31 users (that will become more than 600 in 2013).

Some images of the GUI are reported to give an example of usage. The GUI form in Figure 7 is the one where the user specifies: what kind of document he is inserting (a "Presentazione"), the market (in this case "Insurance") and the business items the document is about (selecting these values using drop down lists that group the business item instances in their subclasses). In the example, the user chooses the instances "User_Experience", "Web_development" and "Digital_media_strategy" from class *Attivitá_professionale*. Finally, the user chooses the language ("Italiano").

In Figure 8, the system presents to the user the list of the current tags that he selected plus the tags that have been automatically added: in this case only the tag "Digital_Marketing_Design" from class *Business_Area*, because all the selected business items refer to this area, as shown in Table II.

Note that in the ontology, Labels have been added to

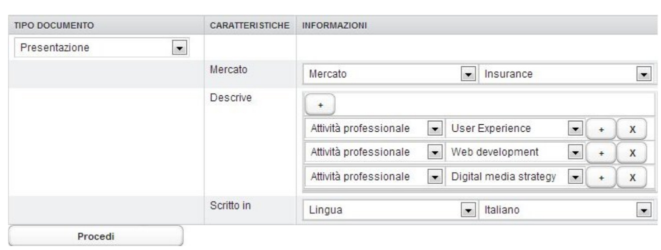

Figure 7. First GUI form for inserting a new document with some first tags. Screen shot.

TAG	CATEGORIA
Insurance	Mercato
User Experience	Elemento d'offerta
Web development	Elemento d'offerta
Digital media strategy	Elemento d'offerta
Italiano	Lingua
Digital_Marketing_Design	Area_business

Figure 8. List of a manually selected tags plus those automatically added. Screen shot.

Table II
Business_Item INSTANCES ASSOCIATED TO THE "DIGITAL_MARKETING_DESIGN" BUSINESS AREA.

Business_Item instances of "Digital_Marketing_Design" business area			
User experience	Movie Design	ADV Campaign	Brand Identity
Digital Media Strategy	Visual Graphic Design	Digital Marketing	Web Development

classes to store the term to be shown in the GUI, because sometime the class names are not "good to be visualized" (for example they contain the "_" character, or are in a different language from the GUI one). For example, in Figure 8 the tags from class *Business_Item* are called "Elemento d'offerta" instead of the standard class name.

Then the system asks the user if he wants to add some more tags, choosing from those connected to the already selected ones. In the example, the user chooses to add more tags starting from the business area "Digital_Marketing_Design". So the system shows the user the possible tags, choosing from the instances related to "Digital_Marketing_Design" (considering the properties with domain *Business_Area*) in the ontology (Figure 9). The user can add some of those tags and then saves the document.

Scegli ulteriori tag da associare al documento

TAG	CATEGORIA
Community	Ambito tecnico
Forum	Ambito tecnico
Portale	Ambito tecnico
Sito web	Ambito tecnico
Sito mobile	Ambito tecnico
Product	Mercato
Financial Services	Mercato

>
<

TAG	CATEGORIA
Insurance	Mercato
User Experience	Elemento d'offerta
Web development	Elemento d'offerta
Digital media strategy	Elemento d'offerta
Italiano	Lingua
Digital_Marketing_Design	Area_business

Figure 9. Possible new tags (left) and already associated tags (right). Screen shot.

In this first prototype of the EC^2M system, the Loader (or Classifier) module, based on Lucene and Jena, is already able to automatically classify a document given as input using a simple version of the algorithm described in [17]. The algorithm that automatically extracts the tags is based on the "term-frequency value" calculated for every possible tag selected from the ontology (the same that the user can choose in the manual process). For every possible tag in the ontology, its synset is read from the ontology (this set is made of the terms that are synonyms or related words, stored in labels in the ontology): then a score for the tag is calculated analyzing the document and searching in it the tag and its synset, using different weights. Lastly, only a subset of the tags is selected to be related to the document, with respect to a threshold.

V. CONCLUSION AND FUTURE WORK

The EC^2M project has been fully tested in 2012 and showed very good results with respect to a standard ECM system: it really helped users from different companies better collaborate, exploiting a semantic classification of their documentation and consequently offering a simpler searching phase and a better support in sharing information, that was impossible to obtain without a similar system. The ontology is now complete and model all the document's types and contents. The notification of new documents to subscribed users is done in quasi real time. Furthermore, with the deployment over the cloud platform, the performances can be enhanced with a new purchase of cloud services: with this architecture and deployment solution the system is really scalable. The system has been adopted by Sempla, and related companies, as their new content management system. Furthermore, from the industrial viewpoint, the system has a good "Return Of Investment" because the adoption of open source technologies and the chance of exploiting the system as a service over the cloud platform (with a pay-per-use solution) allowed a reasonable initial budget and a high scalability in the overall architecture.

Considering the automatic extraction of tag, those with a threshold higher than 0,3 are correctly associated with the document in the 95% of the executed tests: this module will be improved in the next months, but the first results are already promising.

Analyzing the state of the art, many other systems that exploit ontologies to improve knowledge sharing have been presented, dealing with domains that are completely different from ours (for example [18], that is a semantic television content management system based on ontologies) or relying on different architectures (for example [19], where an Ontology Server (OS) component is created to be used in a distributed content management grid system), but the underling problem still remains the same, proving that it is still open and studied. The solution that we adopted is a particular one, where an ad hoc ontology is integrated into a CMS and then deployed over a cloud platform, following the new trends in different research areas, but many other solutions exist, as those described here or previously in Section I. It is impossible to make a precise comparison with the other mentioned systems, because they are commercial and because a completely new system, using the other

 ISBN: 978-1-61208-273-8

approaches, should be developed to be tested and compared, and this solution is not feasible.

In the next months, we are going to complete the EC^{2}M system enhancing the phase of automatic extraction of tags from texts, that is now at a very preliminary stage. Furthermore, we will investigate how dealing with the scenarios where the nodes in the EC^{2}M network use different ontologies to describe and tag their documentation: in this case, the common ontology must be anyway chosen and defined, but ontology matching techniques may be adopted to align the common and private ontologies before tagging and when receiving notifications from the other nodes, to let them keep on using their private ontology but also being able to share documents with common tags.

ACKNOWLEDGMENT

The research described in this paper has been funded by the "EC^{2}M system (Enterprise Cloud Content Management)" Programma Operativo Regionale (POR) project. The first author has been also partially supported by the Progetto di Ricerca di Ateneo (PRA) project "Modellazione e Verifica Automatica per la Sicurezza di Sistemi in "Future Internet"" funded by Genova University.

The authors would like to thank Maurizio Ferraris from Nacon and Viviana Mascardi and Gianna Reggio from DIBRIS that led the industrial and the DIBRIS academic components, respectively, involved in the EC^{2}M project.

REFERENCES

[1] AIIM. Association for Information and Image Management, *What is Enterprise Content Management (ECM)?*, 2010.

[2] The Alfresco Homepage. http://www.alfresco.com/. Retrieved: March, 2013.

[3] R. J. Nagle, *A User's Guide to Plone*, Enfold System Inc., 2010. Retrieved: March, 2013.

[4] The Sensenet homepage. http://www.sensenet.com/. Retrieved: March, 2013.

[5] The smartlogic homepage. http://www.smartlogic.com. Retrieved: March, 2013.

[6] The hdose homepage. http://dose.sourceforge.net/. Retrieved: March, 2013.

[7] The OpenCalais homepage. http://www.opencalais.com/. Retrieved: March, 2013.

[8] The Apache Stanbol homepage. http://stanbol.apache.org/. Retrieved: March, 2013.

[9] R. Garca, J. M. Gimeno, F. Perdrix, R. Gil, M. Oliva, *The Rhizomer Semantic Content Management System*, In M. Lytras, JM. Carroll, E. Damiani, and RD. Tennyson (Eds.), Emerging Technologies and Information Systems for the Knowledge Society, LNCS Volume 5288, 2008, pp. 385-394.

[10] C. Frank, M. Gardoni, *Information content management with shared ontologies - at corporate research centre of EADS*, International Journal of Information Management Volume 25, Issue 1, 2005, pp. 55-70.

[11] D. M. Le and L. Lau, *An Open Architecture for Ontology-Enabled Content Management Systems: A Case Study in Managing Learning Objects*, LNCS, Numb. 4275, 2006, pp. 772-790.

[12] The OWL language overview homepage. http://www.w3.org/TR/owl-features/. Retrieved: March, 2013.

[13] The Protégé homepage. http://protege.stanford.edu/. Retrieved: March, 2013.

[14] T. R. Gruber, *A translation approach to portable ontology specifications*, Knowledge Acquisition 5 (2), 1993, pp. 199-220.

[15] N. F. Noy and D. L. McGuinness, *Ontology Development 101: A Guide to Creating Your First Ontology*, Stanford TR KSL-01-05 and TR SMI-2001-0880, 2001.

[16] The Bibliographic Ontology homepage. http://bibliontology.com/specification. Retrieved: March, 2013.

[17] K. S. Jones, *A statistical interpretation of term specificity and its application in retrieval*, Journal of Documentation, 28, 1972, pp. 11-21.

[18] J. L. Redondo-García and A. Lozano-Tello, *ONTOTV: an ontology-based system for the management of information about television contents*, International Journal of Semantic Computing, Volume 6, 2012, pp. 111-130.

[19] A. Aiello, M. Mango Furnari, A. Massarotti, S. Brandi, V. Caputo, and V. Barone, *An experimental ontology server for an information grid environment*, International Journal of Parallel Program, 34, 6, 2006, pp. 489-508.

