A Knowledge Management System based on Ontologies

Article

Accepted version

P. Ribino, A. Oliveri, G. Lo Re, S. Gaglio

In Proceedings of International Conference on New Trends in Information and Service Science, 2009, pp. 1025-1033

Publisher: IEEE

http://www.computer.org/csdl/proceedings/niss/2009/3687/00/3687b025-abs.html

A Knowledge Management System based on Ontologies

Patrizia Ribino, Antonio Oliveri, Giuseppe Lo Re and Salvatore Gaglio
Dipartimento di Ingegneria Informatica (DINFO)
Università degli Studi di Palermo
Palermo, Italy
Email:{ribino,antonio.oliveri,lore,gaglio}@unipa.it

Abstract

Recently the companies' interest on a correct knowledge management is grown, more than interest on the mere knowledge itself. In the last few years, several projects have been carried out, with the aim of the development of innovative systems capable of collecting and sharing information. This paper proposes a Knowledge Management System, whose main feature is an ontological knowledge representation. The ontological representation of data allows of specializing the reasoning capabilities and of providing ad hoc behaviors. The system has been tested collecting and using data coming from projects and processes typical of ICT companies, and provides a Document Management System and an Expert System to share documents and to plan how to best use firms' knowledge.

1. Introduction

The great amounts of data and documents generated during working activities impose to companies the adoption of new computer-based information systems, capable of enabling the storage of structured data and the automation of the information-processing activities of the organization. Recent studies have drawn attention to problems related to knowledge capitalization and management [1],[2]. During the last two decades ad-hoc frameworks known as *Knowledge Management Systems* (KMS) have been proposed, enabling access and coordination of knowledge assets [3]. KMSs are information systems which manage organizational knowledge increasing the productivity of knowledge operators. This paper proposes an ontology-based knowledge management framework capable of modeling generic offices structure and ICT company's projects, discovering offices' processes to be automatized and company's projects to be reused or developed ad hoc. The system is composed by different modules, such as an expert system for decision support and a document management engine. The main goal is to produce a system for managing data with a separation between knowledge representation and knowledge management; changes in the structure of concepts in the knowledge base do not influence the inference mechanisms and logical reasoning processes.

The paper is structured as follows: section 2 provides a vision of the state of the art of KMSs and expert systems; in section 3 we introduce essential aspects of KMSs, Expert Systems for decision support and Ontologies; in section 4 we present a detailed description of the system; section 5 illustrates the Document Management Service of the KMS; section 6 presents our system implementation. Finally, in section 7, we trace some conclusions.

2. State of the Art

Starting from studies about what knowledge is [4],[5], during last decade researches on Knowledge Management has examined several issues about new strategies, tools and systems in order to organize, store and share data and individuals' expertise; at the same time the most important ICT companies have demonstrated an increasing interest in the development of internal knowledge management instruments, using novel data representation models and involving modern AI techniques in order to promote the dissemination of the knowledge acquired during job activities by different operators. It is easy to identify common features in the KMSs of this researches, most of them using technological infrastructures based on web corporate portals. A web and ontology-based KMS called WAICENT (World Agriculture Information Centre) is in use on the United Nations Food and Agriculture Organization. Using this platform FAO's knowledge about food security and agricultural development becomes widely available to users through the Internet, adopting visualization systems to show information in an immediate way, and decision-support systems in order to improve food security through information use. A document management system is also provided by the KMS, allowing sharing and use FAO's documentation [6]. L. Razmerita et al.[7] proposed a KMS based on the ontological model of the user profile, representing users' preferences, competencies and so on, adopting semantic web techniques. Many other KMSs in literature are designed and customized to satisfy the needs of specific firms. Liping Sui [8] and Maya Daneva [9] studied the benefits of a decision support system within the business management. D.J. Harvey and R.Holdsworth

[10] observed the advantage of using a KMS in the aerospace industry. An interesting study conducted by Chun, Sohn and Granados [11], shows the use of a Knowledge Management System in an industrial engineering company. Through observation of the company's requests, they identified the features of a KMS which were necessary in order for it to be considered a good investment for the firm.

In this article we introduce a prototypal KMS applied to a real case. Exploiting the advantages of the representation of domain concepts through ontologies, two knowledge management mechanisms have been implemented, one related to efficient document management and the other concerning decisional problems facing efficient government.

3. Overview

3.1. Knowledge Management

Knowledge Management (KM) consists of a technique that uses Information Technology tools for the management of information, and its goal is to improve the efficiency of work teams; it studies methods for making knowledge explicit, and sharing a firm's professional expertises and informative resources. There are different classification criteria for KM issues. Alavi and Leidner [3] group the problems of the KM, namely storage, creation, transfer and retrieval issues into four classes. Others researchers studied knowledge problems from different points of views. Verwijs et al. in [12],[13] analyzed different knowledge approaches in business processes, and categorized them as follows:

- **Knowledge storage approach**: knowledge is seen as a product or resource, as something that can be made explicit through codification. Knowledge storage involves issues for obtaining and capturing knowledge from organizational members and/or external sources.
- **Knowledge processes approach**: knowledge as more of a production factor than mere information [14]. Experiences, skills and attitudes are also part of this knowledge. The focus of this approach is on the transfer of knowledge through human interaction. There are many questions about knowledge transfer, one of the most important concerns knowledge flows between provider and knowledge seeker. From the provider's point of view, flow is a selective pull process, but from a seeker's perspective, flow is a selective push process. Thus it is necessary to strike a balance between pull and push processes [15].
- **Learning processes approach**: it involves the relationships between individual and organizational learning. Knowledge management can be seen as assisting organizational learning processes through which new knowledge is created.
- **Intellectual capital approach**: measuring the value of knowledge that resides in an organization [16], this approach focuses on rendering explicit the knowledge of an organization, obtaining the transfer of tacit or implicit knowledge to explicit and accessible knowledge formats. This methodology aims at optimizing the infrastructure of an organization, and includes training, customers relationships, ICT, work organization and so on .

3.2. Knowledge Management Systems

A generic KMS, supporting the creation and storage of knowledge, creates the opportunity to make data, information and knowledge from different sources readily available. KMSs contain both explicit and tacit knowledge. Explicit knowledge, more familiar and easily written down, includes data stored in documents. KMSs can also store tacit knowledge, which is more difficult to express, and includes people's experiences, know-how and expertise. As Takeuchi and Nonaka [5] say "The more important kind of knowledge is tacit knowledge." Tacit knowledge is more difficult to manage than explicit knowledge and can be expressed as "we know more than we can tell". Tacit knowledge, created by experience, is highly personal. The issue of how to better capitalize and disseminate tacit knowledge is one of the actual priorities in Knowledge Management. To realize such goals, a KMS can make use of different technologies such as:

a. *Document based*: technologies for the creation, administration and sharing of different documents (such as doc, pdf, html and so on), managing the explicit knowledge of an organization.

b. *Ontology/Taxonomy based*: technologies using ontologies and classification for knowledge representation. Knowledge concepts are frequently arranged in hierarchical structures, typically related by relationships. Such methodologies act on both explicit and tacit knowledge.

c. *AI based*: using particular inference engines to resolve peculiar domain problems, the framework based on these technologies generally manipulates tacit knowledge (g.e. Knowledge-base system).

3.3. Ontologies

A formal definition of ontology is the following: "*An Ontology is an explicit specification of a conceptualization*" [17]. In computer science the use of the term "*ontology*" is derived from the previous use in philosophy, meaning the study of the "*being*", the fundamental categories of which it is composed and the relationships among them. An ontology tries to formulate an exhaustive and rigorous conceptual scheme of a particular application domain. Generally it is represented through a hierarchical structure which contains all the noteworthy entities, the existing relationships between them, the rules, the axioms and the specific domain constraints. Ontologies give an understandable meaning both to humans and to software agents. Given a domain of interest, the ontology explains the knowledge structure creating a syntax of domain terms, and shares it with all the people interacting with the given domain.

3.4. Knowledge-Based Systems

Knowledge-Based Systems (KBS), a class of AI systems, are able to represent specific domain knowledge and apply it for solving problems through inference processes. The essential elements of a KBS, as figure 1 shows [18] are the following:

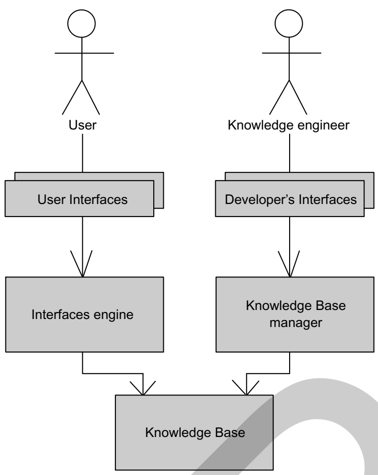

Figure 1. The structure of the Knowledge-based system

- The *Knowledge Base* (KB), the passive component of a KBS, plays a role similar to a database in a traditional informative system;
- The *inference engine*, the elaboration element, uses the KB content to derive new knowledge using reasoning techniques;
- The *Knowledge base manager* manages coherence and consistence of the information stored in the KB.

A particular subclass of KBSs is the rule-based expert system, in which knowledge is captured into a set of rules, each encoding a small piece of the expert's skills. Each rule is an "if-then" statement. An expert system emulates the domain expert in the same conditions.

4. Knowledge management for decision support

In this paper we present a KMS prototype for the automation of government office processes. This prototype is an ontology-based system of knowledge management with the aim of optimizing business processes for creating and managing ICT projects for generic offices. To achieve this, the system implements a document management and an expert system for decision support. The knowledge representation is based on two ontological domain models, the former reproducing the government offices' structure and the latter modeling the concepts of projects developed by an ICT company. The main idea is the separation between knowledge representation and knowledge reasoning, so that the same infrastructure of rules can be used adopting different knowledge bases. Differently from the ontology based KMS reported in section 3, our system provides a general purpose reasoning system, with rules that can adapt their results to different ontological representation of the domain. The proposed system was designed with the following features: specific domain knowledge by building a knowledge base, reasoning ability performed by a rule-based expert system, and finally advanced techniques of document and information retrieval. Figure 2 shows the system architecture.

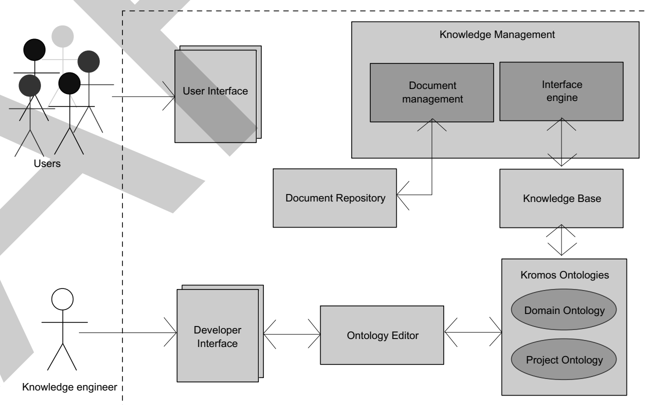

Figure 2. System architecture

The core of the architecture is the KM component, while the data level is composed of a repository to store documents and a KB to maintain domain information. All the components are now presented in detail.

4.1. Knowledge Base

The Knowledge Base represents the knowledge container. KB relations and concepts are described using an ontological structure of instances in order to collect and manage data. The rationale behind our ontology is to provide a minimal but sufficient ontology, suitable for application-domain purpose. Ontologies of the proposed system are performed through Frames [19] and built using Protégé, a free and open source platform developed by Stanford University, that supports frame-based ontologies according to the Open Knowledge Base Connectivity Protocol (OKBC)[20]. In a frame-based model, an ontology is composed by:

- a set of *classes*, hierarchically organized to describe the domain concepts;
- a set of *slots* for the classes, which describes properties and relations between concepts;
- a set of *class instances*, examples of concept with their specific values and properties.

The use of such an ontological model transforms abstract concepts into logical descriptions.

4.1.1. The system Ontologies. Modeling knowledge about the government offices world required some assumptions about its structure and activities, as well as about the nature of the "observer" expected to use, understand, and rely on the model. Our ontology was designed from scratch for the KMS and can be considered a collection of two correlated ontologies, a domain ontology and a projects ontology; in order to keep the ontology easy to understand, only a few concepts from offices' domain and from computer engineering projects are collected. This results in a simplified description of Projects, Processes and Structure of offices and a group of details, attributes and relations. The **Domain Ontology**, a formal representation of the offices' structure and activities is used to characterize the environment in which the system works, and is organized as a set of concepts and relations allowing deduction of new knowledge; it reproduces the logical architecture of offices arranged in levels, each depending on the previous in a pyramidal organization (fig. 3).

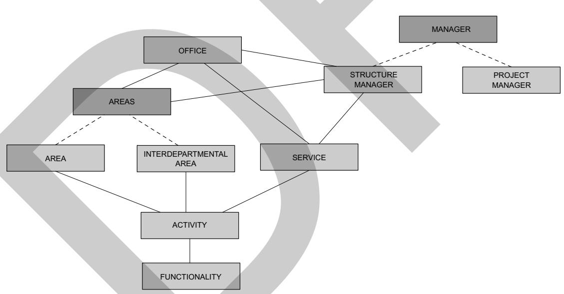

Figure 3. Domain ontology of the KMS platform.

The **Project Ontology** is useful to describe ICT company projects; it maps the structure of the project components containing semi-structured explicit knowledge. During the execution of queries, each component of this ontology is used to get all the elements of the domain in agreement with the query (fig.4).

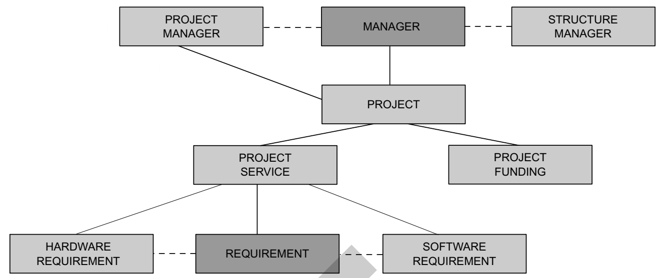

Figure 4. Project ontology of the KMS platform.

4.2. Decision support system

The main goal of an Expert System (ES) for decision support is to assist employees during their activities, finding solutions that usually need the intervention of specifically skilled people. The goal is to incorporate implicit knowledge about the specific field in a computational model. The ES prototype implemented for the platform is a rule-based system developed in Jess [21], that can be used for reasoning in different knowledge base contents, adapting rules to different kinds of domains. Jess, acronym for Java Expert System Shell, is a superset of the CLIPS programming language, developed by E. Friedman-Hill of Sandia National Labs [22]. It provides rule-based programming suitable for automating an expert system, and is often referred to as an expert system shell. Rather than a procedural paradigm, where a single program has a loop that is activated only once, the declarative paradigm used by Jess matches a rule with a single fact specified as its input and processes that fact as its output. When the program is run, the rules engine will activate one rule for each matching fact. Jess can be used to build Java applets, it allows Java functions to be called from Jess code and encapsulates Jess code inside a Java code program, and also creates full applications that use knowledge in the form of declarative rules to draw conclusions and inferences. The ES exploits two different kinds of knowledge: declarative facts, captured by the ontological model, and procedural facts, expressed using rules defined by an expert.

Use cases: Use cases in which our ES can offer support are decisional processes such as:

a) *Project planning process* - During planning of company projects there are many constraints to be considered in order to improve enterprise yields and avoid wasting resources. Planning is a process for the definition of a future goal, the activities to exploit in order to reach that objective, and all the resources to be used to complete these activities. The planning process has to identify business components connected to the real progress of business activities, measuring their impact and benefits, and to analyze the investment policy, producing a Business Plan. Such process is a core activity for the company, because its performances depend on correct planning phase. During this activity, the ES is used as a tool to support the Business Plan editing. Several conditions have to be taken into account by the ES in accordance with certain criteria defined by the human domain expert. For each project, the ES evaluates: the amount of necessary funding for its implementation; the use of human or technical resources; the number of offices that can take advantage by the realization of that project (impact); and so on. After that, the system suggests a plan of activities, with an associated degree of importance, as figure 5 shows.

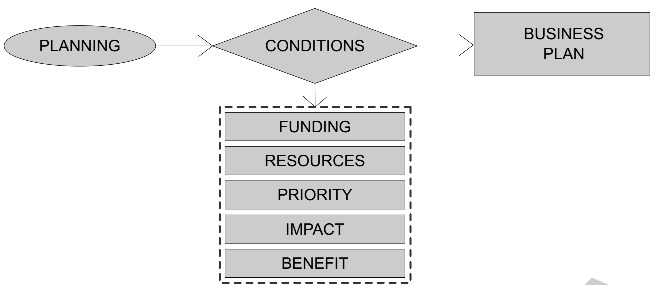

Figure 5. Project planning process

b) *Project management process* - Supervision of the state and progress of projects. This process guides the business management to the attainment of previously planned objectives, showing the differences between them and the results obtained, so that managers can decide and actuate appropriate corrective actions. The progress of each project is defined using specific parameters. The ES can monitor the costs increase, the percentage of utilization of the allocated resources and the time consumed for each work in progress, as figure 6 shows. For instance, if some parameter differs from the value defined during the planning process, the ES suggests some corrective actions.

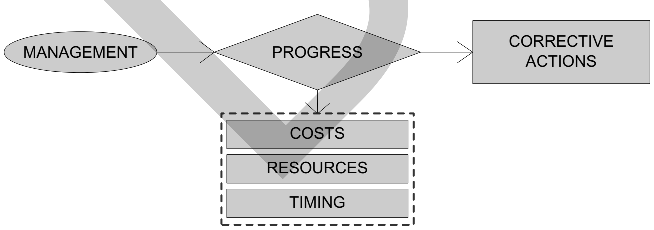

Figure 6. Project management process

c) *Evaluation of previous projects' functionalities reuse* - Analysis of government requirements and already developed projects to find reusable components. The ICT organizations produce a great amount of products to automatize offices' processes. A process is a set of interrelated activities, grouped in phases. Therefore each project is composed by a set of components, each supporting a singular phase of the entire process. Different office processes could have common phases, so that the organization could choose to reuse some components taken from other projects during automation activities, in order to reduce developmental costs. In this case, the ES evaluates some common features between process phases and project components in order to identify which of them could be reused, as shown in figure 7.

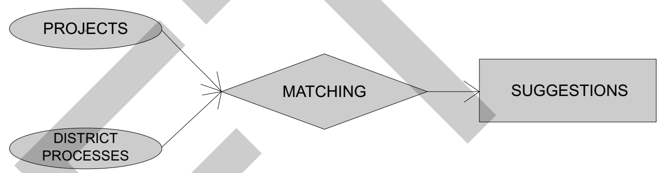

Figure 7. Evaluation of previous projects' functionalities for reuse

4.2.1. Examples of Rules. The code portion below shows a typical ES behavior in the third use case. In this scenario the ES uses the declarative part of the knowledge that expresses concepts of OFFICE and PROJECT and the active part of procedural knowledge emulating the behavior of a human expert.

```
(DEFCLASS office //a general company office
(multislot linked_projects
(comment "projects of the office"
(type INSTANCE))
(single-slot name
(type STRING)
(cardinality 0 1))
(single-slot council_of_competence
(type INSTANCE)
(allowed-classes COUNCIL)
(cardinality 0 1))
(multi-slot comprehend_services
(type INSTANCE)
(allowed-classes SERVICE)
...
(defrule office_processes ?instance
<-(object (is-a OFFICE)
(name ?n&:(call ?*ric* equals
(slot-get"+ off+"name)
(lowcase ?n))))=>(bind $?area
(slot-get (instance-name ?instance)
comprehend_areas
(foreach ?j $?area
(if  (call ?*ric* different
(slot-get (instance-name ?j) name) empty) then...
```

5. Document Management Service

The Document Management Service (DMS) of our KMS is a module capable of pre-processing documents, retrieving data, indexing texts and search engine managing. In ICT companies the volume of documents produced during working activities grows rapidly; as the volume of documents increases, collecting them in traditional forms becomes almost impractical, and searching them without automatic search engines is a great waste of time. Documents contain most of the information about projects, functionalities and so on. Sumiya and Saito proposed [23] a technical instrument to handle different kinds of documents and an easy to use environment to store, collect and search documents. DM can be considered as a set of tasks and processes for collecting, managing and publishing documents and their contents, abstracting and annotating data, enriching KBs with retrieved information and realizing search engines to recover documents from keywords. Our DMS allows the understanding of documents by retrieving information to complete KB instances. With the analysis of technical documents about projects, Project Ontology instances are enriched capturing information like project managers, founding, resources data, keywords to use for research and also about the document file itself, such as type of document, editing, author and so on. Moreover, the Domain Ontology instances are enriched adding relation instances between project and offices that will use the products realized, and which functionalities will be automatized by the project described in the document. Our platform uses Apache Lucene, an Information Retrieval (IR) engine adapted for the insertion, indexing and retrieval of documents in different formats. Lucene is an open-source, high performance, scalable, full-featured text search engine and information retrieval library written in Java and suitable for any application requiring a full text search [24], through which any piece of data convertible to a textual format can be indexed and made searchable [25]. Lucene indexing is constituted by three main phases, converting data to text, analyzing extracted text and saving text to an index. Data in Lucene are stored in the form of Documents. A Lucene Document consists of a collection of Fields, identified by a name and a value, that contain a piece of data queried against or retrieved from the index during search. Searches in Lucene are performed specifying one or more keywords and one or more fields to search within. Search results (in the form of Lucene Documents) are collected within Lucene Hits. Each Document contained in the Hits, has a score value (between 0 and 1) indicating its similarity to the search keys. Since the DMS is based on Lucene, it possesses similar advantages: very fast response time and almost hidden complexity to users. The main functionalities of the DM Service are: pre-processing of documents and their content to obtain a text representation without any lexical or sematic redundance; document indexing to store information about files in an ordered structure to use for the search phase; searching for documents using keywords, calculating the degree of satisfaction of the requirements expressed in the query. These requirements are fulfilled by three modules: *pre-processing*, *indexing* and *searching*.

5.1. Pre-processing module

Pre-processing is a necessary procedure in document management, through which data and information stored in documents in a specific format can be elicited by analyzing and tokenizing content. Organizations generally create and use a great amount of documents that can be stored in different kinds of formats like text files (.txt), document files (.doc, .pdf), web pages (.xml, .html) as figure 8 shows [26]. The analysis of heterogeneous format contents, the removal of meaningless terms and the maintenance of information useful to retrieve and recover documents will depend on the DMS.

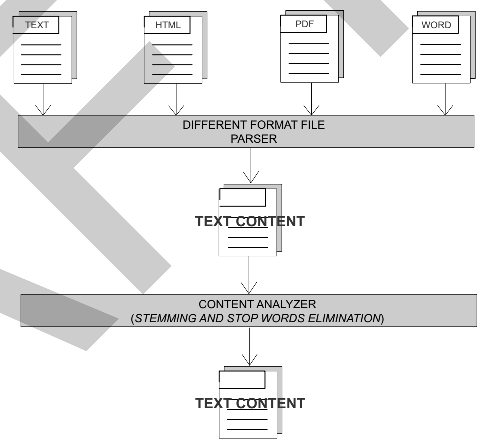

Figure 8. Pre-processing files in DMS

The proposed DMS works in different steps in order to pre-process and organize documents. When an organization's members insert documents to be collected by the KMS, a two-step parsing then occurs: *content elicitation* and *content tokenization*. Content elicitation to withdraw textual content from different kinds of files eliminates irrelevant information, such as typesetting format, and transforms content into a character data stream. Content tokenization breaks the content into words and sentences according to lexical analysis, and transforms the data stream into a set of terms for the subsequent content parsing procedure. After parsing, the pre-process module performs content filtering, to analyze and filter out semantically irrelevant terms in the indexing process, pruning stop words and stemming. We refer to stop words as the terms - such as articles, prepositions, conjunctions, numbers - considered meaningless noise data in content processing. There is another phenomenon in text analysis that can consume memory space and calculation resources; in a common text there can be words such as "tell" in all the variations (told, telling); the system tends to regard those variations as different words with the same word root, even if the semantic meaning is the same. The stemming procedure uses an algorithm to remove the prefix or suffix of words so that the system can interpret them as the same root.

5.2. Indexing module

Apache Lucene was adopted for indexing and document storage, the search interface for querying index and the reading interface to read texts and documents. The fundamental concepts in Apache Lucene are index, document, field and term [27]. An index contains a sequence of documents, which are a sequence of fields. Each field gets tokenized and generates pairs of field name and text tokens called terms.

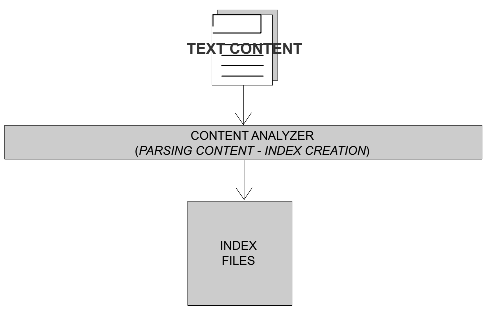

Figure 9. Index creation in DMS

The index stores input in a data structure called inverted index, making efficient use of disk space while allowing quick keyword lookups. The KMS provides a user interface for document insertion, giving users the opportunity to collect them in a unique repository. This activity is absolutely transparent to the user (fig. 9).

5.3. Searching module

After the initial collection of documents and the creation of an index, users need to retrieve documents from the remote file system, seeking them using a search interface. The user interface is also a web-based interface, so that user works as if the interface were a common web search engine, typing keywords, title, project name and so on. The system searches in the indexed documents and retrieves the relevant documents. For this purpose a matching mechanism is used, evaluating the degree to which the document representation satisfies the requirements expressed in the query; Lucene calculates this degree using a set of different factors such as term frequency (number of times a given term appears in the document) and inverse document frequency (obtained dividing the number of all documents by the number of documents containing the term) and retrieves those documents that it believes are relevant, as figure 10 shows. The result is a list of documents, each linked to the related file; when opened, the system displays a reading interface in a *What You See Is What You Get* view, so that the document is integrated in the portal and can be read. In this first release of the software, documents are retrieved on the basis of their content as if they were composed by atomic text entities, that is, a set of different concepts considered here by calculating their frequency in the document, disregarding the document's overall structure.

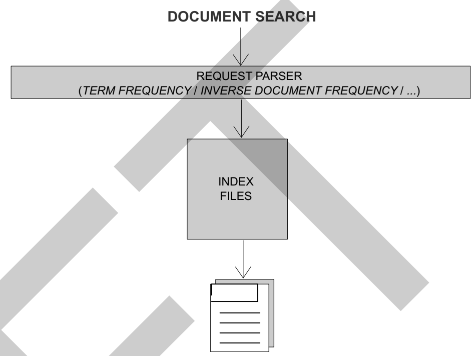

Figure 10. Document search in DMS

6. System Prototype Implementation

The proposed system implements the typical client-server paradigm using JSP (Java Server Page) and Java technologies [28]. JSP plays an important role for web application development. Mainly, JSP allows the separation of dynamic part of web pages from static HTML. Therefore, a JSP is composed by HTML markup in which can be enclosed java code in order to split the page layout from dynamic components. Moreover, with regard to expert system, we can encapsulate Jess into java code so that Java code can interact directly with Jess. Finally, Java can be easily ported to run on various operating systems although we developed our KMS under a Windows XP environment. We used two main server components: Apache Web Server for web page requests and Tomcat Servlet Engine in order to dynamically generate JSP web page. The framework presents different graphical interface through which the users can select different application areas. In our scenario, the system is divided in three different macro-areas: Offices Structure, Process and Project Management. The first allows the user to manage information about the structure of offices. The second enables the user to organize new processes in activities that can be implemented by computer programs. Through the functionalities of Project management area, instead, the user can create complex queries in order to interact with the Expert System. The first two areas involve ontology domain formalization and building processes. The snapshots of a demonstration of our system are shown in Figures 11, 12 and 13.

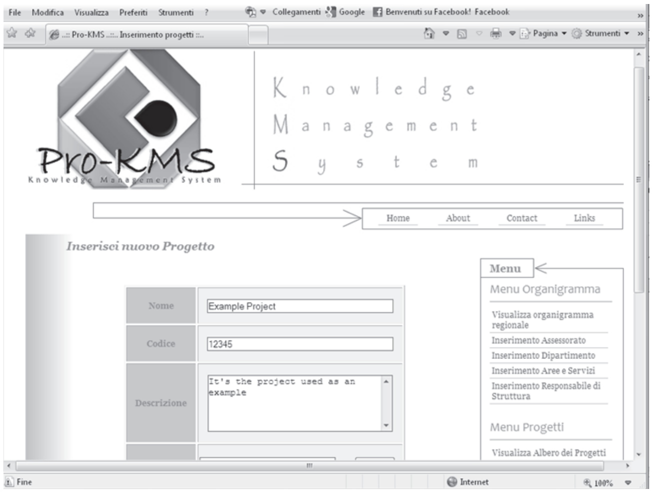

Figure 11. Insert and modify GUI

The user can start in the activities of Knowledge Management, including creation, searching, sharing and so on. From this graphical menu the user can select the above-mentioned different work areas. As previously said, in the first two areas we have implemented some functionalities which allow the creation/change of Knowledge Object (KO), such as domain ontology and projects ontology object respectively. The snapshot shown in Fig. 11 depicts the inserting and editing GUI of projects ontology objects. From this view the user can select an existing KO and fill up new information in order to enrich the Knowledge Base. After that, the KO will be submitted to the server and saved into KB.

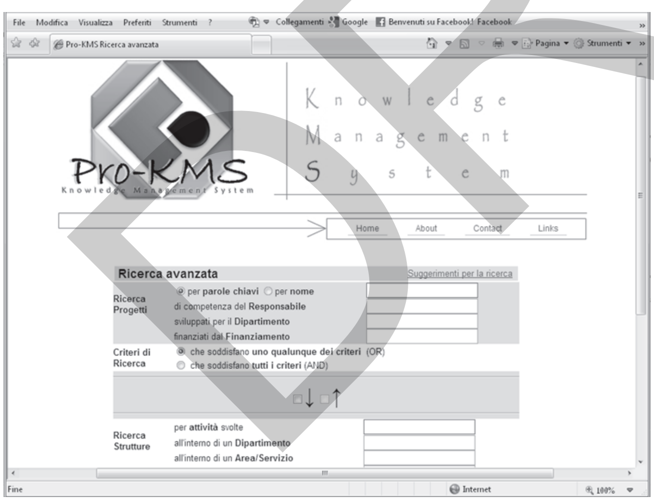

Figure 12. Expert System GUI

The interaction with the expert system occurs through the GUI shown by the snapshot of fig.12. As already mentioned, the ES was implemented using the Jess tool. The expert knowledge and user inputs are stored by rules and ontology instances, respectively. Jess uses the Rete algorithm [29] to match the rules and KO. These rules are composed dynamically according to user requests. This is made possible by enabling different fields and check boxes. Jess outputs are shown on a web page in HTML format. The following snapshot displays the document management interface. This GUI provides documents searching, uploading and consultation.

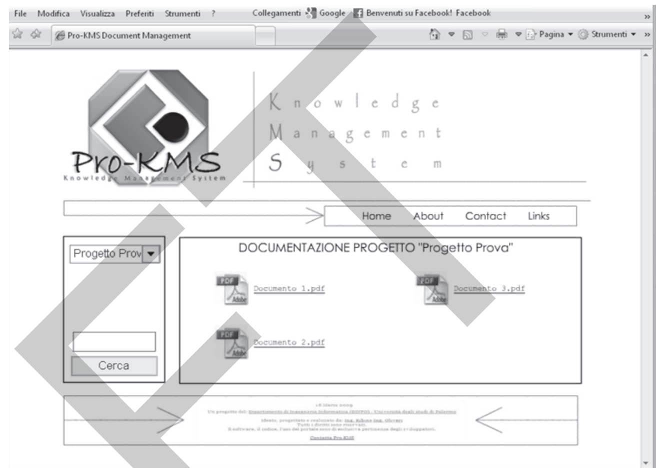

Figure 13. Document management GUI

7. Conclusion

The extremely competitive nature of the new economy, in which knowledge and its efficient exploitation has become a key factor for organizational success, has pressed the organizations to adapt themselves to changing environment. In this regard, Knowledge Management strategies are promoted and several information systems are developed to give support to knowledge processes. That being so, in this article we present a knowledge management solution for an ICT company, which has the aim to improve the growth of organizational knowledge for projects' management. To this aim, we developed a KMS prototype which makes use of a rule-based expert system, which can adapt its results to possible changes in the system ontology, and a document management system in order to improve documents' searching, sharing and discovering. Moreover, the opportunity to use a KMS which models different domains, gives the chance to adapt it to different application contexts and to increase information sharing and reuse.

References

[1] S. Staab, R. Studer, H. Schnurr, and Y. Sure, "Knowledge Processes and Ontologies," *IEEE Intelligent Systems*, pp. 26–34, 2001.

[2] D. O'Leary, "Enterprise Knowledge Management," *Computer*, pp. 54–61, 1998.

[3] M. Alavi and D. Leidner, "Review: Knowledge Management and Knowledge Management Systems: Conceptual Foundations and Research Issues," *Knowledge Management*, 2005.

[4] I. Nonaka, "A Dynamic Theory of Organizational Knowledge Creation," *Knowledge Management: Critical Perspectives on Business and Management*, vol. 5, no. 1, pp. 14–37, 2005.

[5] H. Takeuchi and I. Nonaka, *The knowledge-creating company: How Japanese companies create the dynamics of innovation*. Oxford University Press New York, 1995.

[6] D. O'Leary, "A multilingual knowledge management system: A case study of FAO and WAICENT," *Decision Support Systems*, vol. 45, no. 3, pp. 641–661, 2008.

[7] L. Razmerita, A. Angehrn, and A. Maedche, "Ontology-Based User Modeling for Knowledge Management Systems," *Lecture notes in Computer Science*, pp. 213–217, 2003.

[8] L. Sui, "Decision support systems based on knowledge management," in *Services Systems and Services Management, 2005. Proceedings of ICSSSM'05. 2005 International Conference on*, vol. 2, 2005.

[9] M. Daneva, J. Peneva, R. Rashev, and R. Terzieva, "Knowledge-Based Decision Support System for Competitive Software Audit," in *IEEE International Conference on Systems Man and Cybernetics*, vol. 3, 1995, pp. 1974–1979.

[10] D. Harvey and R. Holdsworth, "Knowledge management in the aerospace industry," in *Professional Communication Conference, 2005. IPCC 2005. Proceedings. International*, 2005, pp. 237–243.

[11] M. Chun, K. Sohn, P. Arling, and N. Granados, "Systems Theory and Knowledge Management Systems: The Case of Pratt-Whitney Rocketdyne," in *Hawaii International Conference on System Sciences, Proceedings of the 41st Annual*, 2008, pp. 336–336.

[12] J. Dijkstra, C. Verwijs, D. de Wit, E. Heeren, and R. Bakker, "Knowledge management for the extended enterprise," in *Proceedings of the 5 thInternational Conference on Concurrent Enterprising, The Hague, Netherlands*, pp. 15–17.

[13] C. Verwijs, "Dijkstra, J., Heeren, E., & de Wit, D., 1999," *Kennismaken met kennismanagement*.

[14] T. Davenport and J. Glaser, "Just-in-time delivery comes to knowledge management," *Harvard business review*, vol. 80, no. 7, pp. 107–111, 2002.

[15] D. Holtshouse, "Knowledge Research Issues," *California Management Review*, vol. 40, pp. 277–280, 1998.

[16] T. Stewart, *The Wealth of Knowledge: Intellectual Capital and the Twenty-first Century Organization*. Doubleday New York, NY, USA, 2001.

[17] T. Gruber, "A translation approach to portable ontology specifications," *Knowledge Acquisition*, vol. 5, pp. 199–199, 1993.

[18] K. M. Hangos, R. Lakner, and M. Gerzson, *Intelligent Control Systems: An Introduction with Examples*. Norwell, MA, USA: Kluwer Academic Publishers, 2002.

[19] M. Minsky, "A Framework for Representing Knowledge," 1974.

[20] V. Chaudhri, A. Farquhar, R. Fikes, P. Karp, and J. Rice, "OKBC: a programmatic foundation for knowledge base interoperability," in *Proceedings of the fifteenth national/tenth conference on Artificial intelligence/Innovative applications of artificial intelligence table of contents*. American Association for Artificial Intelligence Menlo Park, CA, USA, 1998, pp. 600–607.

[21] E. Hill, *Jess in Action: Java Rule-Based Systems*. Manning Publications Co. Greenwich, CT, USA, 2003.

[22] J. Giarratano and G. Riley, *Expert systems: principles and programming*. PWS Publishing Co. Boston, MA, USA, 1994.

[23] S. Sumiya and T. Saito, "Development of a multimedia document management system for cooperative work environment," *Computer Software and Applications Conference, 1992. COMPSAC '92. Proceedings., Sixteenth Annual International*, pp. 346–355, Sep 1992.

[24] O. Gospodnetic and E. Hatcher, *Lucene in Action*. Manning, 2005.

[25] G. Pirro and D. Talia, "An approach to Ontology Mapping based on the Lucene search engine library," in *Proceedings of the 18th International Conference on Database and Expert Systems Applications (DEXA 2007)-Volume 00*. IEEE Computer Society Washington, DC, USA, 2007, pp. 407–411.

[26] D. Zhou and K. Xie, "Lucene Search Engine," *Jisuanji Gongcheng/ Computer Engineering*, vol. 33, no. 18, 2007.

[27] M. Bennett, "Contrasting relational and full-text engines. NIE (New Idea Engineering) Enterprise Search Newsletter 2 (9), article 1," 2004.

[28] K. Avedal, A. Halberstadt, D. Ayers, T. Briggs, C. Burnham, R. Haynes, and S. Zeiger, *Professional JSP*. Wrox Press Ltd. Birmingham, UK, UK, 2000.

[29] C. Forgy, "Rete: A fast algorithm for the many pattern/-many object pattern match problem," *IEEE Computer Society Reprint Collection*, pp. 324–341, 1991.