Proceedings of the International MultiConference of Engineers and Computer Scientists 2012 Vol I,
IMECS 2012, March 14 - 16, 2012, Hong Kong

# SemKnow: A Multi-Agent Platform to Manage Distributed Knowledge by using Ontologies

Davy Monticolo, Inaya Lahoud, Eric Bonjour, Frédéric Demoly

***Abstract*—Nowadays new product development involves different types of actors (technician, managers, board of directors) which must be able to share knowledge, experiences and work together efficiently. Each actor has a professional specialty and uses one or several software tools (CAO, project management tools, PLM tools ...) dedicated to her specific activities. Each of these software tools produces different information sources (databases, XML files, text files) which are distributed through the enterprise network. In this paper, we present the design of a multi-agent software architecture that allows the capitalization of distributed and heterogeneous knowledge. We then describe how the agents handle knowledge through ontologies and build semantic queries.**

***Index Terms*—Multi Agent System, Architecture, Distributed Knowledge Management, Ontology**

## I. Introduction

Current competitiveness has led companies to a fast product renewal coupled with lower costs. Today, manufacturers are creating more and more efficient products while meeting shorter deadlines in order to satisfy customer needs and sales.

These companies have to break into new markets by showing how creative they are to grow more profitable. Such creativity requires an optimized organization mastery, a control of the industrial process and the development of a 'learning company' in which getting knowledge benefits ongoing projects. Learning within the company has now become the best way to be competitive. Learning is not only about improving, but also a way to start a new 'learning culture' in which every co-worker, every team and the whole company will be able to optimize their capacities by sharing their knowledge and their know-how. Thus the industrial interest in methodologies and tools enabling capitalization and management of distributed and heterogeneous knowledge has been growing stronger. This paper describes the design of a MAS platform that shall manage knowledge coming from different information sources. The first part briefly describes the problematic of the management of distributed and heterogeneous knowledge. The second part will explain the design rationale of the SemKnow MAS platform. The last section focuses on our current work on how the agents can manage the problem of knowledge extraction from different software tool databases by using domain ontologies.

D.M.. Author works in the Polytechnical Institute of Lorraine (INPL), 8, rue Bastien Lepage, 54 000 Nancy France, (corresponding author to provide phone: 383 193 249; e-mail: davy.monticolo@ensgsi.inpl-nancy.fr).

I.L., FD Authors are with Unversity of Technology (UTBM) France. (e-mail: inaya.lahoud@utbm.fr,frederic.demoly@utbm.fr).

E. B. is also professor in the Polytechnical Institute of Lorraine (INPL). (email : e-mail: eric.bonjour@ensgsi.inpl-nancy.fr).

## II. The Agent paradigm used in Knowledge Engineering

### A. MAS used in knowledge engineering

The aim of knowledge engineering is to gather, study, organize and represent knowledge. Multi-agent systems have already proved their efficiency to support such tasks. Klusch made a list of the services that a multi-agent system can offer in a knowledge management approach [10]:

- search, acquire, analyse and classify knowledge coming from various information sources;
- Give information to human and computing networks once usable knowledge is ready to be consulted;
- Negotiate on knowledge integration or exclusion into the system;
- Give explanation to the quality and reliability related to the integrated knowledge;
- Learn progressively all along the knowledge management process;

Such services are mostly implemented to create two MAS categories devoted to knowledge management. The first MAS type is based upon an agent cooperation to solve complicated problems related to knowledge types. Some of these SMAs were created as complementary tools in information management (workflow, ontologies, information research systems and so on) to design platforms like FRODO [1], CoMMA [11], Edamok [6], or KRAFT [14]. All these works have focused on the 'Multi-Agent Information System' or MAIS.

The second MAS type gathers management assistant agents depending on the actors' needs. In this range, agents are expected to be flexible, pro-active and reactive regarding the user requirements [9],[10].

The new trend of using MAS in knowledge engineering is to associate agents to knowledge structures based on ontologies. This association allows MAS to support the knowledge management process but the issues of knowledge distribution and ontology consistency with MAS have not been solved yet. The next section will present the related work concerning agent approaches using ontologies in knowledge engineering.

### B. Ontologies to support the knowledge modelling

Ontology is an object of Artificial Intelligence that has become a mature powerful conceptual tool of Knowledge Modelling [5]. It provides a coherent base to build on, and a shared reference to align with, in the form of a consensual conceptual vocabulary on which one can build descriptions and communication acts.

Knowledge that is created in engineering projects needs to be defined precisely in order to be useful in an information system. Ontology provides a vocabulary and a semantic that enable the processing of knowledge related to a specific domain. Ontology is a set of items and their specific meanings. It gives definitions and indicates how concepts are connected to each other. These connections form a structure on the defined domain and clarify the possible meanings of the items [21]. Therefore, a domain ontology includes the specific concepts of a given domain. It describes the entities, properties and the way they can be related to each other. These ontologies are meant to be re-used in the same domain, in new but similar applications. These ontologies are said to be contextual [17] when the concept properties evolve according to the situation.

### C. Interests of the Ontologies in MAS

The idea of using domain ontologies in an agent system aims at reusing pieces of the domain Knowledge to lead agents to share their information. Indeed in a MAS, several agents interact or work together to carry out common goals [20]. The coordination between agents depends on the process and knowledge they use to achieve their global goals. The domain ontology provides a section of the knowledge world that is essential for the agent to carry out its tasks [8].

Some research works like Buccafurri [4] and Wooldridge [22] use the ontology to give to the agents an internal representation of both interests and behaviour of their associated human users. Other works use ontology to help agents to choose the most promising agents to be contacted for knowledge-sharing goals [8], [5]. Generally, these systems have been designed to prevent the agents from having access to the ontologies of other agents; they ensure an individualistic view of agents' societies. This is the viewpoint of most of the so-called BDI approaches [18][12]. Another interesting approach that has been adopted to design MAS is related to the agent community, where agents automatically build their ontologies by observing the users' actions [2]. Indeed, the agents are able to automatically extract logical rules that represent the user behaviour and/or causal implications among events due to the definition of the user interests described with the ontology.

In addition, Guerin [13] and Singh [19] propose to design their MAS in adopting a ''social'' view of agent communities, where it is assumed that the ontology of each agent is, even partially, accessible for each other agent. In the next section, we will propose the architecture of a MAS using a common ontology for all agents. Then, we will present the mechanism of knowledge distribution between agents based on a semantic analysis.

## III. Overview of SemKnow (Semantic & Knowledge)

In this section we present models that enable to build the knowledge management system dedicated to heterogeneous and distributed information management during engineering projects.

### A. Agents and their knowledge worlds

In a knowledge management system, agents are in a complex information world. To handle and transform information into knowledge, they have to identify it by using models, extract it from different sources, annotate it by respecting a knowledge structure, store it, update it and share it with the users.

We use the OWL-Lite (Web Ontology Language) to build ontologies used in SemKnow (Semantic & Knowledge). OWL-Lite provides a simple triple model based on a XML syntax to describe information and their relations. Figure 1 shows an annotation example built by the agents.

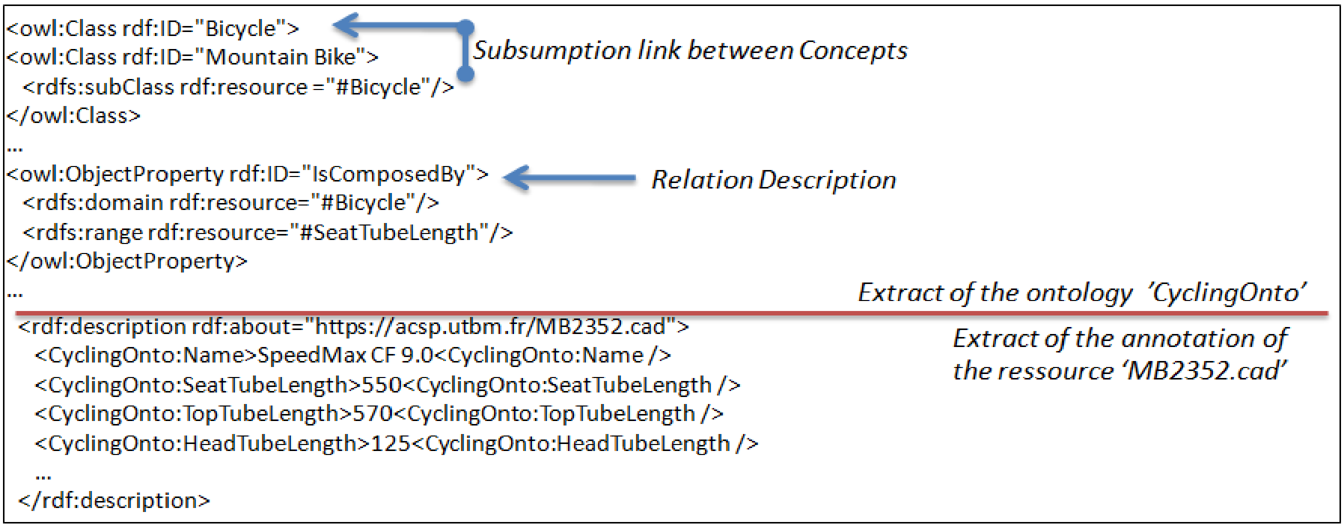


Fig. 1. Example of an annotation build by the agents

The SemKnow system uses several ontologies built by the professional actors. Each ontology describes a domain (project management, mechanical design, Ergonomics, etc.). The agents are connected to several information systems (e-groupware, project management platform, CAD system etc…) and extract the information to these sources by transforming the ontologies into SQL queries. Thus SemKnow manages the heterogeneous and distributed knowledge with RDF annotations based on several shared ontologies. SemKnow is not a library of indexing documents

ISBN: 978-988-19251-1-4
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

but a knowledge base with annotations describing information sources with their organization contexts. The SemKnow has to support the knowledge sharing for the users' organization (project team). The problem is to handle the knowledge of the organization and to ensure the distribution of the relevant knowledge to each agent (either human or artificial). In the following section, we present the SemKnow architecture and the mechanisms of knowledge sharing among the agents.

*B. SemKnow Architecture*

Knowledge agents are a part of cognitive and intelligent agents. They constitute a coupled network of agents that worked together to achieve the same objective i.e. to support the knowledge management process by providing full range of functionalities like extracting, annotating, storing, updating and sharing knowledge [16].

The MAS architecture is a structure of an agent network with different types of agents and different relationships between them [11], [22]. The SemKnow architecture starts from the highest level of abstraction with the description of agent societies and goes down to the description of the roles, interactions and responsibilities of the agents.

The proposed approach to design a MAS is based on an organizational approach like the A.G.R model used in AALAADIN, OPERA and methodologies like GAIA or TROPOS or RIOCC. Thus the SemKnow architecture is tackled as a human society in terms or role, skill and relationships.

The main objective of the SemKnow system is to manage heterogeneous and distributed knowledge coming from different information sources and used by professional actors. The second objective is to permanently evaluate this knowledge in order to delete obsolete knowledge. The third objective is to assist users in the reuse of knowledge by proposing a decision support. Considering these objectives we have defined four main functionalities for the system:

- To allow users to describe their knowledge domain with a semantic approach (i.e. a characterization of concepts and their relations);
- To extract knowledge from different information sources;
- To update and validate the knowledge base with the users in order to avoid broadcasting wrong information;
- To assist the user in the reuse of knowledge.

Figure 2 gives an overview of the SemKnow architecture.

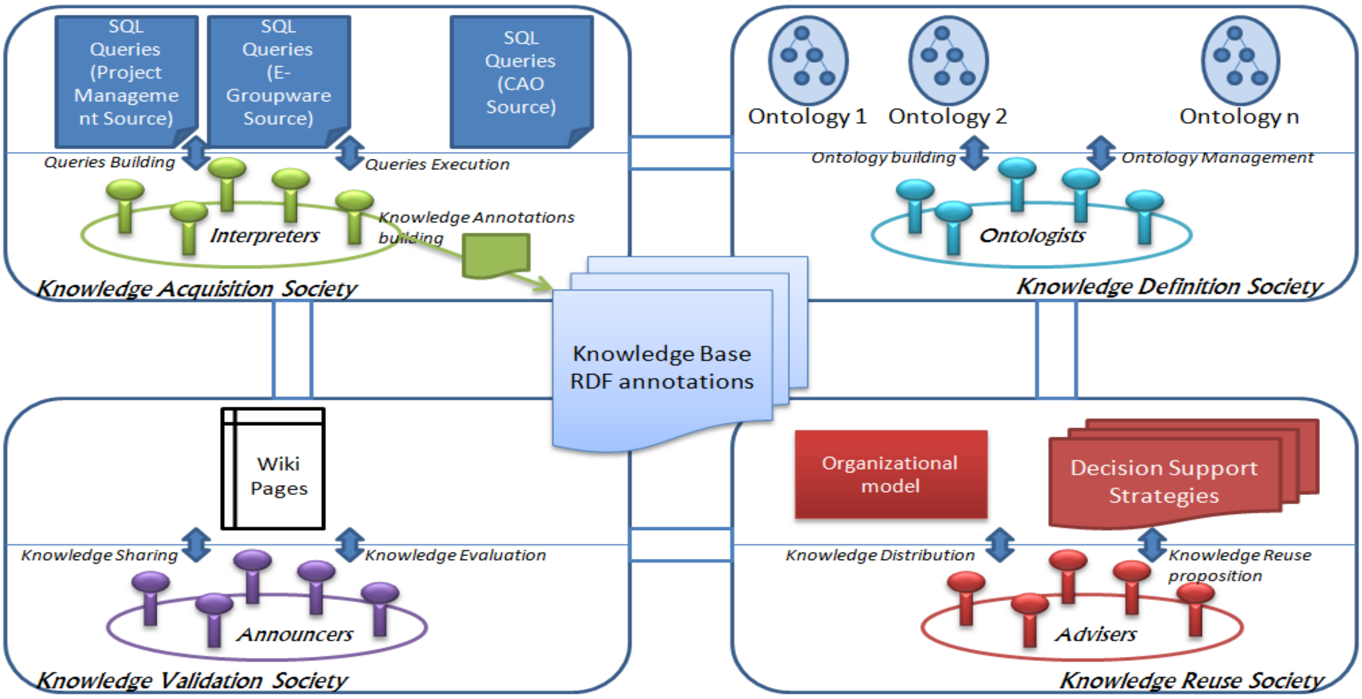


Fig. 2. The SemKnow Architecture

In the high level of abstraction we have four agent societies supporting the four functionalities of the SemKnow platform.

The first society of agents, called "*ontologists*", manages the different ontologies (knowledge models) build by the users. The proposed interface looks like protégé 2000 and the agents help the professional actors to build domain ontologies. The "ontologists" generate ontologies in the OWL-Lite format. There is one ontologist agent for each ontology. The Ontologists also manage the whole ontologies by ensuring the consistency (ontology alignment process and research of similar concepts) between the different ontologies.

The second society of agents is the "*Interpreters*". Those agents extract knowledge by transforming ontologies in query models using the language SQL. Afterwards, they request the databases of different information sources (E-Groupware, Project Management, CAO, etc.) to extract knowledge. There is one agent for each information source. The interpreters annotate knowledge by giving an organizational context (creator's role, information source, name of the project, etc.). The interpreters build the knowledge base with RDF files.

The third society of agents, called the "Announcers", aims

ISBN: 978-988-19251-1-4
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

at evaluating, validating and updating knowledge by using a semantic Wiki. The users can consult knowledge by reading Wiki pages and can modify, approve or reject it. The reader can find more details about this process in [15].

The fourth society of agents is the "*Advisers*". Those agents aim at providing a decision-making support for the users. There is one adviser for each professional actor. The advisers use an organizational model to monitor the actors' actions. This model describes the users' roles, collaborations and activities during a project. Thus, by requesting the knowledge base i.e. the annotations (Knowledge and organizational context), the advisers can alert and propose to users knowledge that has been already stored for a similar past activity, a similar role and a similar information system.

We are going to focus on the work of the *Interpreters* society i.e. how the agents extract knowledge from the different information sources by using the domain ontologies and transformation rules.

## IV KNOWLEDGE EXTRACTION BY THE INTERPRETERS SOCIETY

To extract knowledge from the database of business applications the interpreters have to apply an algorithm to extract and translate information of databases according to the OWL ontologies. Hierarchical information is extracted from matches found in the relational dataset. The agents apply some corresponding rules to detect similarity between the structure of the databases tables and the ontologies. The tables 1 and 2 present the correspondence between database components/properties and ontology concepts/relationships.

Table 1: Component/Concept correspondence

| Database component | OWL concept |
|---|---|
| Table | Class |
| Column | Functional Propoerty |
| Row | OWL individual |
| Column Metadata: | OWL property restriction: |
| Data Type | -AllValues From restriction |
| Mandatory/Not nullable | -Cardinality () Restriction |
| Nullable | -maxCardinality() Restriction |

Table 2: Property/Relationship correspondence

| Database property | OWL relationship |
|---|---|
| NOT NULL | owl:minCardinality rdf:datatype=''& xsd:Int''1/ |
| UNIQUE | owl:InversFunctionalProperty |
| CHECK | owl:hasValue |
| FOREIGN KEY | owl:objectProperty |

We have experimented transformation rules with four different information sources (a e-Groupware, a project management system, a risk analysis system and a CAO platform). The Interpreters apply 11 transformation rules to go from the ontology to the SQL model. These rules are described below:

***R1*** A class is a table

***R2*** If there is an association which is surrounded by the cardinality * on both sides, we search the primary keys that correspond to these 2 tables. Once we found it, we store the name of this table in the list of tables and the primary-foreign key in the relationship table.

***R3*** If there is an association which is surrounded by a * on the first side and a 0 or 1 on the other side we look for the primary key of the table next to *. This key is added as foreign key in the table next to 0 or 1

***R4*** Store the name of tables in a table

***R5*** Search the primary and foreign keys of the tables in the list (the relations between them) and store these relations in a table.

***R6*** Forming conditions: we consider the range of each DatatypeProperty

a. If it is an attribute positiveInteger then the condition is > 0
b. If it is a DataRange which includes a list, so attribute in [value1, value2 ...]
c. If there are restrictions on DatatypeProperty and it is not a cardinality restriction as HasValue, so the condition is: check attribute = value

***R7*** Inverse functional property is the "Distinct" constraint

***R8*** Required property mean that the fields are not null (not 0 or not = "")

***R9*** Symmetric property is a recursive table R10 Store conditions in a table.

***R11*** Build the query

a. Select all DatatypeProperty by the first letter of the domain table. e.g.: p.nom, and followed by a comma
b. From all tables stored in the table (R4) followed by the first letter as naming the table by a letter
c. Where all the relationships stored in a table to make the connections between the tables + all conditions stored in the table separated by "and"

We illustrate the mechanism used by the interpreters for the knowledge extraction with an ontology concerning the cycling domain.

The cycling ontology describes the cycling world. The ontology defines a vocabulary and a semantic to structure, organize, detail all the characteristics of a bike, all the roles of the professional actors during the development project of a new bike and all the processes used to develop and industrialize a bicycle.

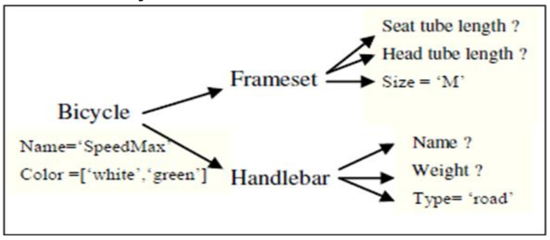


Fig. 3. The SemKnow Architecture

This ontology is transformed into an SQL query which will return information from business applications. For example if business experts want to know how to design a 'SpeedMax CF' bike size M so the *Interpreters* agents transform the cycling ontology (figure 4) in specifying these details. The cycling ontology will be transformed into an SQL query such as the example below. The information returned with this query becomes knowledge and stored in RDF files as shown in Figure 1 in the annotation part.

ISBN: 978-988-19251-1-4
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

```
SELECT Handlebar.TypeHandlebar, Frameset.Size,
Frameset.SeatTubeLength, Frameset.HeadTubeAngle,
Bicycle.Color, Bicycle.Name, ...
 FROM Handlebar, Frameset, Bicycle, ...
 WHERE Bicycle.IdHandlebar = Handlebar.IdHandlebar
 AND Bicycle.IdFrameset = Frameset.IdFrameset
 AND ...
AND Bicycle.Name like 'SpeedMax%'
 AND Bicycle.Color in ('White','Green')
 AND Frameset.Size ='M'
 AND Handlebar.TypeHandlebar != 'road'
```

Fig. 4. Extract of the generated SQL query

By using those rules, we have succeeded in extracting 58% of the concepts defined in the ontologies. It is more than previous work like in [2] but we have to more improve this result. Indeed it depends on the structure of databases. If the database has not relevant relations we obtain a multitude of results which are not relevant.
The loss of semantics is due to the fact that some of the concepts and relations defined in the ontology have no equivalence in the database (table or relations). In the SemKnow platform, the professional actors build their own ontology from their professional expertise. We observed that when actors had a good acquaintance with the professional software tool he/she used, then he/she properly defined ontology with concepts close from the database structure and the agents obtained good knowledge extraction results.

## IV. Conclusion

This paper has presented the architecture of the SemKnow platform with four agent societies which cover the knowledge management process. It has focused on the knowledge extraction that is carried out by the *Interpreters* agents. The next step of this research will be to improve the obtained results according to two ways; the first will be to find new transformation rules to decrease the loss of semantics and the second concerns the development of new models used by the ontologists agent society, to better assist the human experts in the building of domain ontologies that with similar concepts that they use when they work with their professional software tools.

## References

[1] A. Abecker, A. Bernardi, and L. van Elst, 2003, Agent technology for distributed organizational memories. In Proceedings of the 5th International Conference On Enterprise Information Systems, Vol. 2, pages 3–10

[2] Astrova, I., Korda, N., and Kalja, A.2007. Storing OWL Ontologies in SQL Relational Databases. World Academy of Science, Engineering and Technology 29

[3] T. Atanasasova, H-J. Nern, M. Hamalainen, H. Nour Eldin, Distributed heterogeneous knowledge database for control system design: Multiagent development and support, ISIC, 12p., Patras, Greece, 2000.

[4] F. Buccafurri, D. Rosaci, G.M.L. Sarne´ , D. Ursino, An agent-based hierarchical clustering approach for e-commerce environments, in: Proceedings of the Third E-Commerce and Web Technologies (EC-Web 2002), Aix-en-Provence, France, Lecture Notes in Computer Science, Springer, Berlin, 2002, pp. 115–118.

[5] G. Beydoun, N. Tran, G. Low and B. Henderson-Sellers, Foundations of Ontology-Based MAS Methodologies, Agent-Oriented Information Systems III, SpringerLink, Volume 3529/2006

[6] M. Bonifacio, P. Bouquet, and P. Traverso. Enabling distributed knowledge management. Managerial and technological implications. Novatica and Informatik/Informatique, 3(1):23– 29, 2002

[7] R. Braga, C. Werner, M. Mattoso, Odyssey-Search: A multi-agent system for component information search and retrieval, The Journal of Systems and Software, vol79 (2006) p204–215.

[8] M. C. Bravo, J. Pérez, V. J. Sosa, A. Montes, G. Reyes, Ontology Support for Communicating Agents in Negotiation Process, International Conference on Hybrid Intelligent System, 12p. 2005

[9] P.A Champin, Y. Prié, A. Mille, MUSETTE: Modelling USEs and Tasks for Tracing Experience, Proc. From structured cases to unstructured problem solving episodes - WS 5 of ICCBR'03, Trondheim (NO) , NTNU, Trondheim (NO) , pp279-286, June 2003

[10] M. Klusch, (Ed.): Intelligent Information Agents: Agent-based Information Discovery and Management in the Internet, Springer, 1999.

[11] F. Gandon, L. Berthelot , Dieng-Kuntz R.,"A Multi-Agent Platform for a Corporate Semantic Web",AAMAS 2002, 6th International Conference on Autonomous Agents, 5th International Conference on Multi-Agents Systems, 9th International Workshop on Agent Theories Architectures and Languages Eds Castelfranchi C., Johnson W.L., p. 1025-1032, July 15-19, 2002, Bologna, Italy.

[12] N. Griffiths, M. Luck, Cooperative plan selection through trust, in: Proceedings of the Ninth European Workshop on Modelling Autonomous Agents in a Multi-Agent World: Multi-Agent System Engineering (MAAMAW-99), vol. 1647, Springer, Berlin, 1999, pp. 162–174.

[13] F. Guerin, J. Pitt, Denotational semantics for agent communication language, in: AGENTS '01: Proceedings of the Fifth International Conference on Autonomous Agents, ACM Press, 2001, pp. 497–504

[14] J. Kyengwhan, Jung-Jin Yang, Knowledge Description Model for MAS Utilizing Distributed Ontology Repositories, in Agent Computing and Multi-Agent Systems, SpringerLink, Volume 4088/2006

[15] D. Monticolo, S. Gomes "Collaborative Knowledge Evaluation with a Semantic Wiki: WikiDesign", in the International Journal of e-Collaboration, Volume 7, Issue 3. Copyright © 2011. 12 pages.

[16] M. R.Lee, T. T. Chen, " Revealing Research Themes and Trends in KnowledgeManagement: from 1995 to 2010", in Knowledge-Based Systems, December 2011.

[17] Pouquet P., *C-owl: Contextualizing ontologies*. In Proceedings of the 2nd International Semantic Web Conference (ISWC2003), 20-23 October 2003, Sundial Resort, Sanibel Island, Florida, USA., Oct 2003

[18] A. Rao, M. Georgeff, Decision procedures of bdi logics, J. Logic Comput. 8 (3) (1998).

[19] B. Singh. *Interconnected Roles* (IR): A Coordination Model. Technical Report CT-084-92, MCC, (1992).

[20] K. Sycara, "*In-Context Information Management through Adaptive Collaboration of Intelligent Agents*", p53-77 in the book "Intelligent Information Agent: Agent-Based Information Discovery and Management on the Internet", Matthias Klusch, Springer, 1999

[21] Uschold M., "Building ontologies: towards an Unified Methodology. In proceedings of the 16th conference of the British Computer Society Specialist Group and Expert Systems, Cambridge UK, 1996.

[22] M. Wooldridge, Reasoning About Rational Agents, MIT Press, Cambridge, MA, 2000.

ISBN: 978-988-19251-1-4
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)