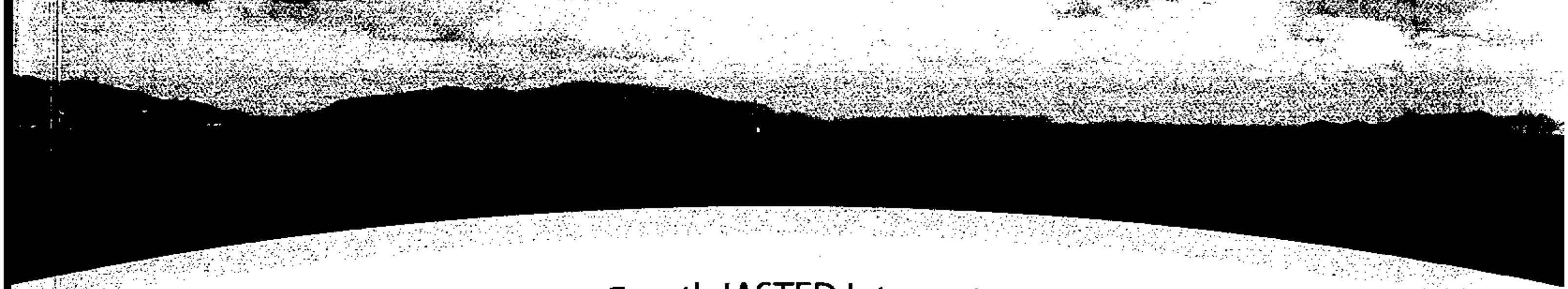
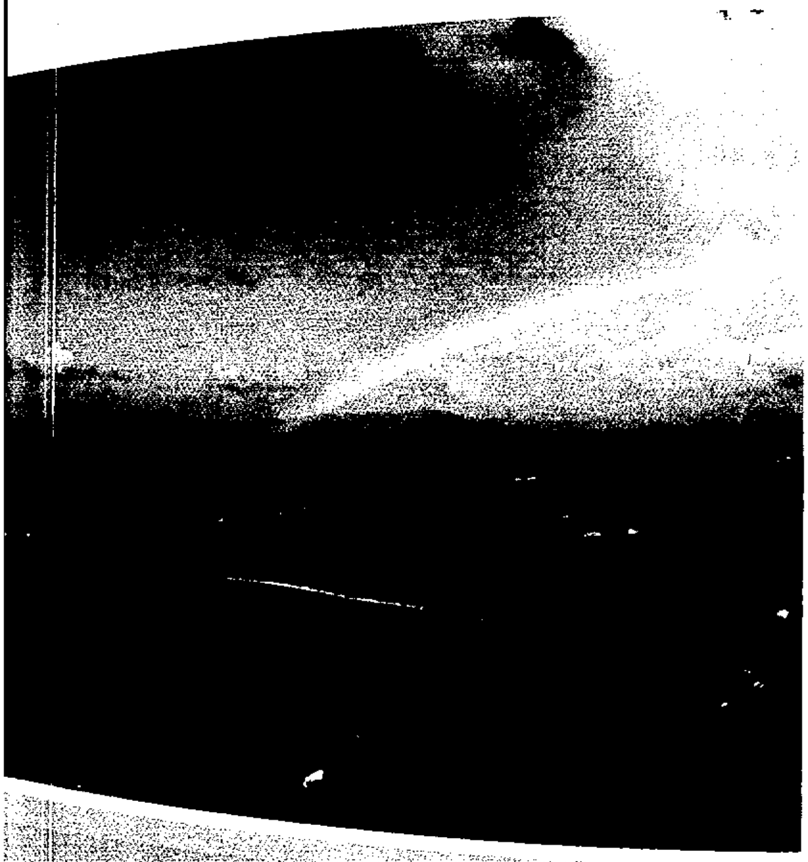
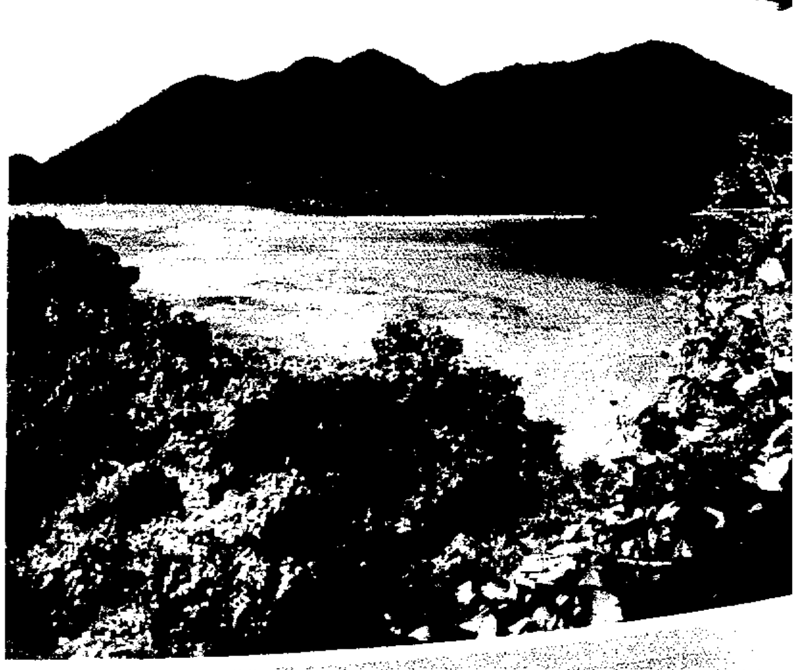
A Publication of the International Association of Science and Technology for Development



Proceedings of the Fourth IASTED International Conference on

Knowledge Sharing and Collaborative Engineering





Editors: M. Boumedine
C. Touzet

November 29 – December 1, 20
St. Thomas, US Virgin Is

ISBN: 0-88986-609-0

ACTA Press Anaheim | Calgary | Zurich

Proceedings of the Fourth IASTED International Conference on **Knowledge Sharing and Collaborative Engineering**, held November 29 – December 1, 2006, St. Thomas, US Virgin Islands.

SPONSOR

The International Association of Science and Technology for Development (IASTED)

- Technical Committee on Information Systems
- Technical Committee on Telecommunications
- Technical Committee on the Web, Internet, and Multimedia
- Technical Committee on Management

IN COOPERATION WITH

CAIG
Caribbean Artificial Intelligence Group

For each IASTED conference, the following review process is used to ensure the highest level of academic content. Each full manuscript submission is peer reviewed by a minimum of two separate reviewers on the International Program Committee/Additional Reviewers list. The review results are then compiled. If there are conflicting reviews, the paper is sent to a third reviewer.

ACTA Press
P.O. Box 5124
Anaheim, CA 92814–5124
USA

ACTA Press
B6, Suite #101, 2509 Dieppe Ave SW
Calgary, Alberta T3E 7J9
Canada

ACTA Press
P.O. Box 354
CH–8053 Zurich
Switzerland

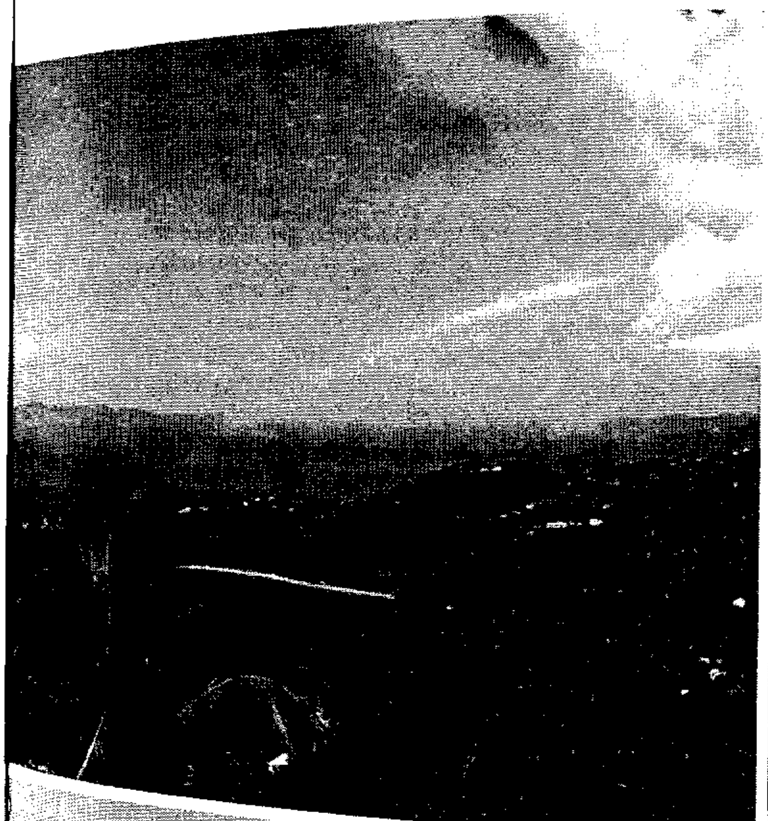
Publication Code: 532

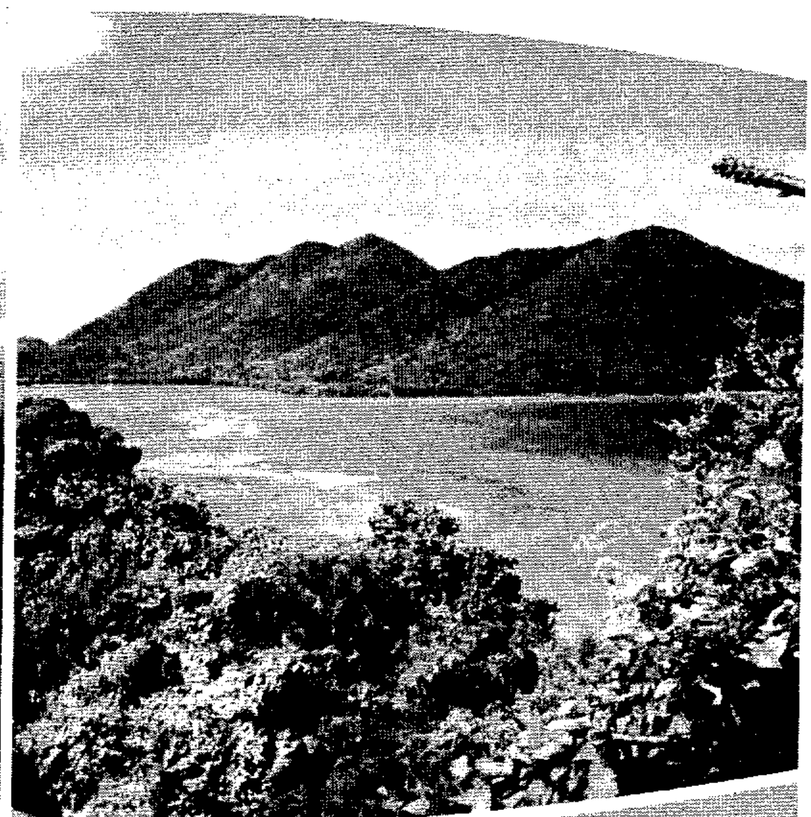
IASTED

A Publication of the International Association
of Science and Technology for Development

Proceedings of the Fourth IASTED International Conference on

Knowledge Sharing and Collaborative Engineering





Editors: M. Boumedine
C. Touzet

November 29 - December 1, 2006
St. Thomas, US Virgin Islands

ISBN: 0-88986-609-0

ACTA Press Anaheim | Calgary | Zurich

TABLE OF CONTENTS
KSCE 2006

Proceedings of the Fourth IASTED International Conference
Knowledge Sharing and Collaborative Engineering
November 29 – December 1, 2006, St. Thomas, US Virgin Islands
ISBN Hardcopy: 0-88986-609-0 / CD: 0-88986-611-2

MAPROK: A MULTI-AGENT ARCHITECTURE TO DEVELOP KM SYSTEMS

Juan Pablo Soto, Aurora Vizcaíno, Javier Portillo and Mario Piattini
Alarcos Research Group
Information Systems and Technologies Department, UCLM-Soluziona Research and Development Institute
University of Castilla - La Mancha
Paseo de la Universidad, 4-13071
Ciudad Real, Spain
jpsoto@proyectos.inf-cr.uclm.es, aurora.vizcaino@uclm.es,
javier.portillo@alu.uclm.es, mario.piattini@uclm.es

ABSTRACT
The goal of this paper is to illustrate that the usage of software agents can be beneficial in the development of systems to manage knowledge. Considering that knowledge is distributed throughout different sources and has a heterogeneous nature, agent technology may be interestingly applied to this problem, since multi-agent systems are often distributed and agents have proactive and reactive features which are very useful for Knowledge Management Systems (KMS). This paper describes an approach that allows us to build robust and adaptable KM systems by using a multi-agent architecture. This architecture is called MAPROK (a Multi-agent Architecture to PROcess Knowledge) which uses specialized autonomous agents for specific services and allows agents to interact in order to support the main knowledge processes.

KEY WORDS
Software Agents, Knowledge Management Systems.

1. Introduction

In the last few decades, Knowledge Management (KM) has captured the attention of businesses as being one of the most promising ways to attain success in this information era [1]. We are currently living in a knowledge economy where the principal economic resource businesses have to offer their customers is knowledge. A shorter life-cycle of products, globalization, and strategic alliances between companies demand a deeper and more systematic organizational knowledge management [2]. In order to assist organizations to perform this task, systems have been designed to manage organizational knowledge. These are called Knowledge Management Systems (KMS), defined by Alavi and Leidner [3], as an IT-based system developed to support/enhance the processes of knowledge creation, storage/retrieval, transfer, and application.

Due to the importance of knowledge management, tools which support some of the tasks related to KM have been developed. Different techniques are used to implement these tools. One of them, which is proving to be quite useful, is that of intelligent agents [4]. Software agent technology can monitor and coordinate events, meetings and disseminate information [5]. Furthermore, agents are proactive in the sense that they can take the initiative and achieve their own goals. The autonomous behaviour of the agents is critical to the goal of this research since agents can act on behalf of their users in carrying out difficult and often time-consuming tasks that the employees have to perform when using a KM system. Consequently, agent systems are expected to become more efficient with time since the agents learn from their previous mistakes and successes [6]. Most agents today employ some type of artificial intelligence technique to assist the users with their computer-related tasks, such as reading e-mails, maintaining a calendar, and filtering information [7].

Because of these advantages different architectures have been proposed to support activities related to KM [8]. Some agent-based architectures have even been designed to help in the development of KMS. However, most of them are focused on a particular domain and can only be used under specific circumstances. What is more, they do not take into account the cycles of knowledge in order to use knowledge management in the system itself. For these reasons, in this paper we propose a generic multi-agent architecture to develop KM systems. The agents used in this architecture have been designed to support the different phases of a knowledge life cycle with the goal of fostering the creation, storing and dissemination of knowledge.

This paper is structured as follows. Section two describes related works based on multi-agent architectures. In section three our proposal is explained and we also define the roles of the agents and tasks of each in charge of supporting the knowledge process. Finally, conclusions and future work are outlined in section four.

2. Related works

Traditional knowledge management systems have received certain criticism as they are often implanted in the company thus overloading employees with extra work, as the employees have to introduce information into the KMS and worry about updating this information. One proposal to avoid this overloading is to add software agents to perform this task in place of the employees.

The benefits of applying agent technology to knowledge management include distributed system architecture, easy interaction, resource management, reactivity to changes, interoperation between heterogeneous systems, and intelligent decision making. The set of KM tasks or applications in which an agent can assist is virtually unlimited, for instance:

- CoMMA [8] project, Corporate Memory Management through Agents, combines emergent technologies (e.g. XML, machine learning) allowing users to exploit an organizational memory.
- Wang et al. in [9] propose a multi-agent architecture to provide support to cooperative activities.
- Another area where agents have been used is in recommendation systems and expert location. These systems are used to help people find information or experts which can assist them in their daily work [10].
- In [11] a multi-agent system is proposed for knowledge sharing in a system designed to advise good programming practice.

Besides these works we found others which focused on document classification [12, 13], mailing list management [14], data mining [15], corporate memories [16], etc.

Most previous works have been developed without considering how knowledge flows through a company and how it is stored in different knowledge sources. Because of this, they often support only one knowledge task without taking into account that knowledge management implies giving support to the different processes and activities, for instance those which form the SECI model (see [17]) and that guarantee an increase of knowledge.

3. Multi-agent Architecture

This section describes our architecture for developing KM systems and how we intend to apply it jointly with data sources to process knowledge.

Figure 1 shows a schematic representation of this architecture. The architecture has been built by using a three layer model. The functionality of those layers can be summarized as:

- **Local Agent Layer**. This layer has two agents: the *Interface Agent* and the *Personal Agent*. The *Interface Agent* acts as an effective bridge between the user and the rest of the agents. Such agents actively assist a user in operating an interactive interface. The best-known examples of interface agents are intelligent tutoring systems and context-sensitive help systems. [18] is a good example of a system in which the user may operate the interface with complete disregard for the agent, but if called upon, the agent may also display suggestions, or perform direct-manipulation actions upon objects in the displayed interface, based on input collected implicitly from the user. In our knowledge management system, the Interface Agent is proposed to help users to search for useful information related to their activities and to give recommendations about employees' expertise or knowledge sources that can be consulted. Furthermore, it interacts with the user receiving user specifications and delivering results. The second agent is the *Personal Agent* which obtains user profiles and information relevant to users' knowledge that helps to determine the expertise level and knowledge that each person has or that a person may need. There are several ways in which agents can be trained to better understand user preferences by using computational intelligence techniques, such as using evolutionary computing systems, neural networks, adaptive fuzzy logic and expert systems, etc.
- **Knowledge creation layer**. This layer provides a Knowledge Agency. It consists of the Constructor Agent, the Captor Agent, the Searcher Agent, the Disseminator Agent and the Maintenance Agent. This agency is in charge of giving support to the KM processes proposed in [19]. The knowledge created is stored in a knowledge repository.

  We shall now explain each of the agents that form this agency. The *Captor Agent* is responsible for collecting the information (data, models, experience, etc) from the different knowledge sources. It executes a proactive monitoring process to identify the information and experiences generated during the interaction between the user and groupware tools (email, consulted web pages, chats, etc.). In order to accomplish this, the Captor Agent uses two ontologies: the knowledge ontology which defines the knowledge to be taken into account in a domain and the source ontology which defines where each type of knowledge might be found. The *Constructor Agent* is in charge of giving an appropriate electronic format to the experiences obtained so that they can be stored in a knowledge base to aid retrieval. Storing knowledge helps to reduce dependency on key employees because at least some of their expert knowledge has been retained or made explicit. In addition, when knowledge is stored, it can be made available to all employees, providing them with a reference as to how processes must be performed, and how they have been performed in the past. Moreover, the Constructor Agent compares the new information with old knowledge that has been stored previously and decides whether to delete it and add new knowledge or to combine both of them. The *Searcher Agent* is in charge of giving recommendations or suggestions with the goal of helping users to perform their tasks by reusing lessons already learnt. This agent could be implemented with different retrieval techniques. Several solutions for information extraction have been proposed. Some of the most popular solutions, which have actually been implemented, are related to the semantic web [20]. Others use XML-based specifications [21] and ontologies [22] to represent, the information stored in the repository in a coherent way. Another agent is the *Disseminator Agent*, which must detect the group of people, or communities who generate and use similar information: for example, in the software domain, the people who maintain the same product or those who use the same programming language. Therefore, this agent fosters the idea of a community of practice in which each person shares knowledge and learns thanks to the knowledge of the other community members [23]. An appropriate type of knowledge management linked to communities of practice helps to improve the organization's performance [24]. Disseminated information may be of different types; it may be information linked to the company's philosophy or specific information about a determined process. Finally, the Disseminator agent needs to know exactly what kind of work each member of the organization is in charge of and the knowledge flows linked to their jobs. In order to do this, the Disseminator Agent contacts the *Personal Agent* (explained in the previous layer) which gives him information about the profiles of the user. Comparing this stage with the SECI model we can say the Disseminator Agent fosters the socialization process since it puts people who demand similar knowledge in touch and once in contact they can share their experience,. thus increasing their tacit knowledge. Finally, there is also a *Maintenance Agent*. The main purpose of this agent is to keep the knowledge stored in the knowledgebase updated. Therefore, information that is not often used is considered by the Maintenance Agent as information to be possibly eliminated.

- **Data sources layer**. This layer is composed of data repositories (email, web, documents, and other sources). Repositories of collected and combined data sources become part of our architecture. It is important to keep such repositories updated with constant contributions because the knowledge must be shared. We are applying data mining technology [25] to manage the information that comes from the diverse repositories. Data mining involves the semiautomatic discovery of interesting knowledge, such as patterns, associations, changes, anomalies and significant structures, from large amounts of data stored in databases and other information repositories. Data mining functions can be implemented using a variety of technologies, such as database-oriented techniques, machine learning, statistical techniques, and another artificial intelligence methods [26].

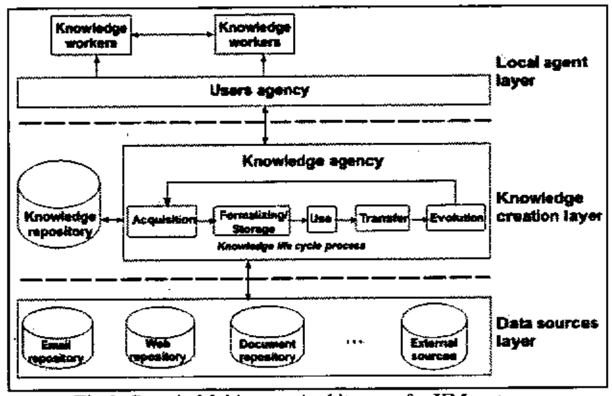


Fig 1. Generic Multi-agent Architecture for KM systems

The different layers are interconnected through the agencies explained here (Figure 1).

In order to carry out the analysis and design of the agents involved we have followed a methodology called INGENIAS [27] which provides meta-models to define Multi-agent Systems, and support tools to generate them. Using meta-models facilitates the development of systems enormously, since they are oriented towards visual representations of concrete aspects of the system.

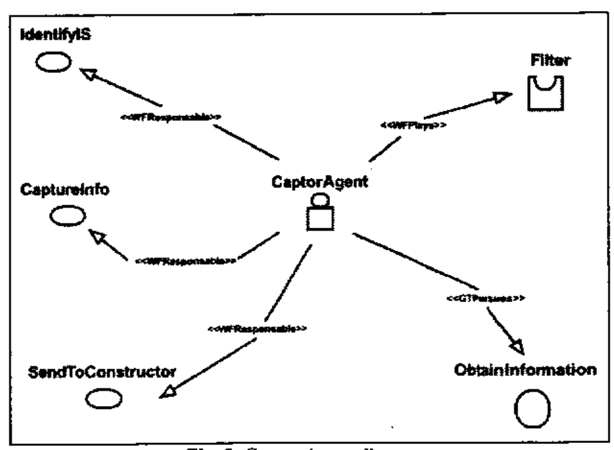


Fig 2. Captor Agent diagram

Below, we are going to explain two agent meta-model diagrams which describe the roles and tasks of each one of the agents that supports the knowledge process. Roles represent the position of an agent in a society and the responsibilities and activities assigned to this position and expected to be fulfilled by the other. The agents that form the local agent agency are omitted due to space limitations.

Figure 2 depicts the role and tasks performed by the Captor Agent. The Captor Agent role (Dredge) must gather information from the system and user's experience that can be transformed into knowledge with the goal of being used in future projects. Next, we describe each one of the tasks developed by this agent.

CaptureInfo: The agent must capture information.
IdentifyIS: This task consists of identifying available information sources in the system.
SendToConstructor: Once information has been captured the agent must send it to the Constructor Agent.

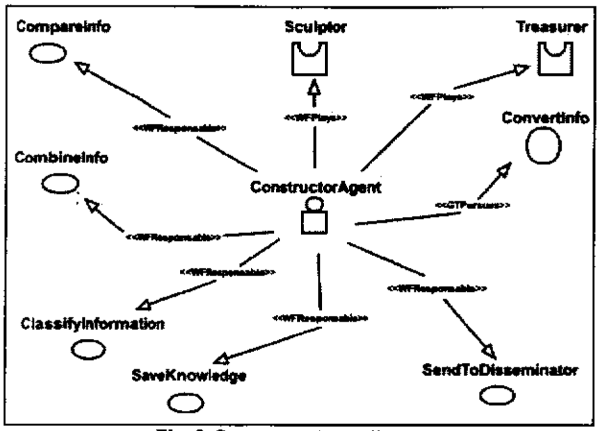


Fig. 3. Constructor Agent diagram.

Another important agent that supports the knowledge creating process is the Constructor Agent. Figure 3 shows the role and tasks performed by this agent. The Constructor Agent is in charge of giving an appropriate electronic format to the experiences obtained so that they can be stored in a knowledge base. The tasks developed by Constructor Agent are:

CompareInfo: In this task the Constructor Agent compares the new information with old knowledge previously stored and decides whether to delete it or to add new knowledge.
CombineInfo: This task is in charge of combining the new information with the previously stored knowledge. In this way, the combination process of the SECI model is carried out, producing new knowledge resulting in the merging of explicit knowledge with new explicit knowledge.
ClassifyInformation: This task is in charge of classifying the information received by the Captor Agent (for instance: models, structures, files, diagrams, etc.).
SendToDisseminator: This task is in charge of sending knowledge to the Disseminator Agent.
SaveKnowledge: This task is in charge of storing the new knowledge in the knowledge base.

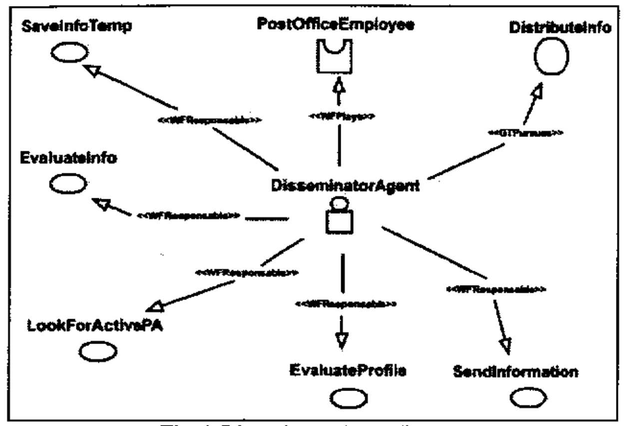


Fig. 4. Disseminator Agent diagram.

The Disseminator Agent diagram (see Figure 4) is composed of the following tasks:

SaveInfoTemp: this task temporally stores the new knowledge received by the Constructor Agent.
EvaluateProfiles: Once the user profile has been identified, the Disseminator Agent needs evaluate it to determine users' necessities.
PostOfficeEmployee: This task is in charge of distributing the information.
EvaluateInfo: This task is focused on evaluating information received in order to relate it to different user's profiles.
SendInformation: This task is in charge of sending the information and knowledge required by the users to allow them to perform their tasks.
LookForActivePersonalAgents: Personal agents can be distributed in different nodes, so this agent must identify all the active Personal Agents available in the system.

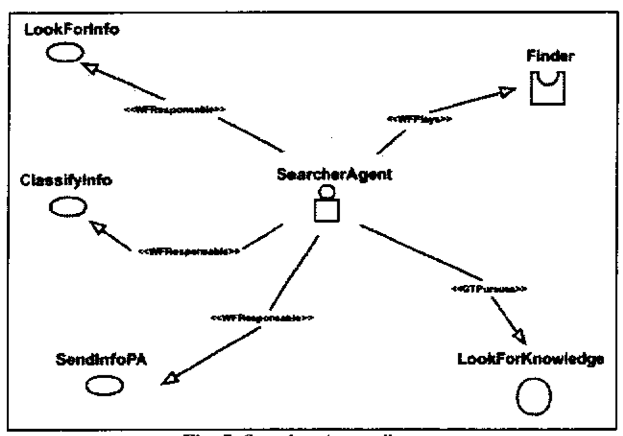


Fig. 5. Searcher Agent diagram

Another agent that support the knowledge life cycle is the Searcher Agent. The goal of this agent is to foster the internalization process of the SECI model, since the employees have the opportunity of acquiring new knowledge by using the information that this agent suggests.

The Searcher Agents diagram (Figure 5) is composed of the following tasks:

LookForInfo. This task is in charge of searching for information that may be useful for the users.
ClassifyInfo. This task is in charge of classifying the information found in the knowledge base.
SendInfoToPA. This task is in charge of sending the knowledge found in the knowledge base to the Personal Agent.

Finally, the Maintenance Agent diagram is described (see Figure 6). The main purpose of this agent is to keep the knowledge stored in the knowledge base updated.

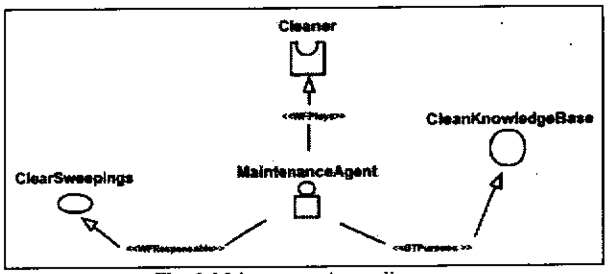


Fig. 6. Maintenance Agent diagram

The platform that we are using to develop the architecture is JADE since it is FIPA compliant and is currently one of those most used . Moreover, JADE has been successfully used in the development of other systems in the domain of knowledge management [28-30].

We are presently also studying JADEx in order to see how easy it would be to migrate to this new platform.

5. Conclusion and further work

This work describes a multi-agent architecture for the creation of knowledge management systems. The main contribution of this paper is the design of a multi-agent architecture which considers the main processes of a knowledge life cycle such as creating, maintaining, sharing and distributing knowledge. Therefore, it is a generic multi-agent architecture to be taken into account when developing knowledge management systems. The usage of the INGENIAS models helps possible users of this architecture to understand how the different agents work and what their roles are.

In future work we aim to compare the implementation of a KMS based on our architecture with developments using other architectures. This evaluation will, without any doubt, help us to improve our proposal.

Acknowledgements

This work is partially supported by the MAS (TIC2003-02737-C02-02), ENIGMAS (PIB-05-058), ESFINGE (TIN2006-15175-C05-05) project, Junta de Comunidades de Castilla-La Mancha, Consejería de Educación y Ciencia, Spain and CONACYT (México) under grant of the scholarship 206147/229687 provided to the first author.

References

[1] D. Malone, *Knowledge Management: A Model for Organizational Learning*. International Journal of Accounting Information Systems, 2002. Vol. 3: pp. 111-123.
[2] M. Ale, O. Chiotti, and M.R. Galli, *Agent-Supported Ontology-Based Knowledge Management Systems*. Proceedings of ASIS (Simposium Argentino de Sistemas de Información) and JAIIO (33° Jornadas Argentinas de Informática e Investigación Operativa), 2004.
[3] M. Alavi and D.E. Leidner, *Knowledge Management and Knowledge Management Systems: Conceptual Foundations and Research Issues*. MIS Quarterly, 2001. Vol. 25(1): pp. 107-136.
[4] L. van-Elst, V. Dignum, and A. Abecker, *Agent-Mediated Knowledge Management*. International Simposium (AMKM'03), 2003, Stanford, CA, USA: Springer, pp. 1-30.
[5] S. Balasubramanian, R. Brennan, and D. Norrie, *An Architecture for Metamorphic Control of Holonic Manufacturing Systems*. Computers in Industry, 2001, Vol. 46(1): pp. 13-31.
[6] P. Maes, *Agents that reduce work and information overload*. Communications of the ACM, 1994, Vol. 37(7): pp. 31-40.
[7] Jensen, J. *Using an Intelligent Agent to Enhance Search Engine Performance*. 2002, Available from: http://www.firstmonday.dk/issues/issue2_3/jansen/index.html.
[8] F. Gandon, *Multi-Agent System to Support Exploiting an XML-based Corporate Memory*. Proceedings PAKM 2000, Basel, Switzerland.
[9] A. Wang, C. Reidar, and L. Chunnian, *A Multi-Agent Architecture for Cooperative Software Engineering*. Proceedings of the Eleventh International Conference on Software Engineering and Knowledge Engineering, 1999.
[10] R. Crowder, G. Hughes, and W. Hall, *An Agent Based Approach to Finding Expertise in the Engineering Design Environment*. Proceedings of the 14th International Conference on Engineering Design, 2003.
[11] S. Mercer, and S. Greenwood, *A Multi-Agent Architecture for Knowledge Sharing*. in *Proceedings of the Sixteenth European Meeting on Cybernetics and Systems Research*. 2001.
[12] J. Novak and M. Wurst, M. Fleischmann and W. Strauss, *Discovering, Visualizing and Sharing Knowledge through Personalized Learning Knowledge Maps*.

Proceedings of the Agents-Mediated Knowledge Management (AMKM), 2003.

[13] J. Yang, P. Pai, V. Honovar and L. Miller, *Mobile Intelligent Agents for Document Classification and Retrieval: A Machine Learning Approach.* Proceedings of the Fourteenth European Meeting on Cybernetics and Systems Research (EMCSR'98). 1998. Vienna, Austria, pp. 707-712.

[14] E. Moreale and S. Watt, *An Agent-Based Approach to Mailing List Knowledge Management.* Proceedings of the Agent-Mediated Knowledge Management (AMKM), 2003.

[15] C. Giannella, R. Bhargava and H. Kargupta, *Multi-agent Systems and Distributed Data Mining.* In Cooperative Information Agents VIII: 8th International Workshop (CIA-04), 2004, Erfurt, Germany, Springer-Verlag, Vol. 3191: pp. 1-15.

[16] T. Cao and F. Gandon, *Integrating External Sources in a Corporate Semantic Web Managed by a Multi-Agent System.* Proceedings of the Agent-Mediated Knowledge Management (AMKM), 2003.

[17] I. Nonaka, *A dinamic theory of organizational knowledge creation.* Organization Science, 1994, Vol. 5: pp. 14-37.

[18] T. Selker, *COACH: A Teaching Agent That Learns.* Communications of the ACM, 1994. Vol. 37(2): pp. 92-99.

[19] J.P. Soto, A. Vizcaíno and M. Piattini, *A Multiagent Architecture to Process Knowledge.* In *IADIS International Conference WWW/Internet 2006.* Murcia Spain.

[20] T. Berners-Lee, J. Hendler and O. Lassila, *The Semantic Web.* Scientific American, 2001.

[21] J.M. Bremer and M. Gertz, *XQuery/IR: Integrating XML document and data retrieval.* Proceedings of the 5th International Workshop on the Web and Databases (WebDB), 2002.

[22] T. Gruber, *Towards Principles for the Design of Ontologies Used for Knowledge Sharing.* International Journal of Human-Computer Studies, 1995, Vol. 43(5/6): pp. 907-928.

[23] E. Wenger, *Communities of Practice: Learning Meaning, and Identity.* 1998, Cambridge U.K., Cambridge University Press.

[24] E. Lesser and J. Storck, *Communities of Practice and Organizational Performance.* IBM Systems Journal, Germany, 2001, Vol. 40: pp 831-841.

[25] M.S. Chen, J. Han and P.S. Yu, *Data mining: An Overview from a Database Perspective.* IEEE Transactions on Knowledge and Data Engineering, 1996. Vol. 8(6): pp. 866-883.

[26] S.C. Hui and G. Jha, *Data minig for Customer Service Support.* Information and Management, 2000. Vol. 38: pp. 1-13.

[27] J. Pavón, and J.J. Gómez-Sanz. *Agent Oriented Software Engineering with INGENIAS.* in *Multi-Agent Systems and Applications III: CEEMAS 2003 (LNAI).* 2003. Prague, Czech Republic, Springer.

[28] A. Abecker, A. Bernardi and L. van-Elst, *Agent Technology for Distributed Organizational Memories: The Frodo Project.* Proceedings of 5th International Conference on Enterprise Information Systems, 2003, Angers, France.

[29] F. Bergenti, A. Poggi and G. Rimassa, *Agent Architectures and Interaction Protocols for Corporate Memory Management Systems.* Proceedings of the 14th International Conference on Artificial Intelligence. Workshop on Knowledge Management and Organizational Memories, 2000, pp. 39-47.

[30] F. Gandon, *A Multi-Agent Architecture for Distributed Corporate Memoriès.* In AT2AI-3, 3rd International Symposium From Agent Theory to Agent Implementation in European Meeting on Cybernetics and Systems Research, 2002, pp 623-628.