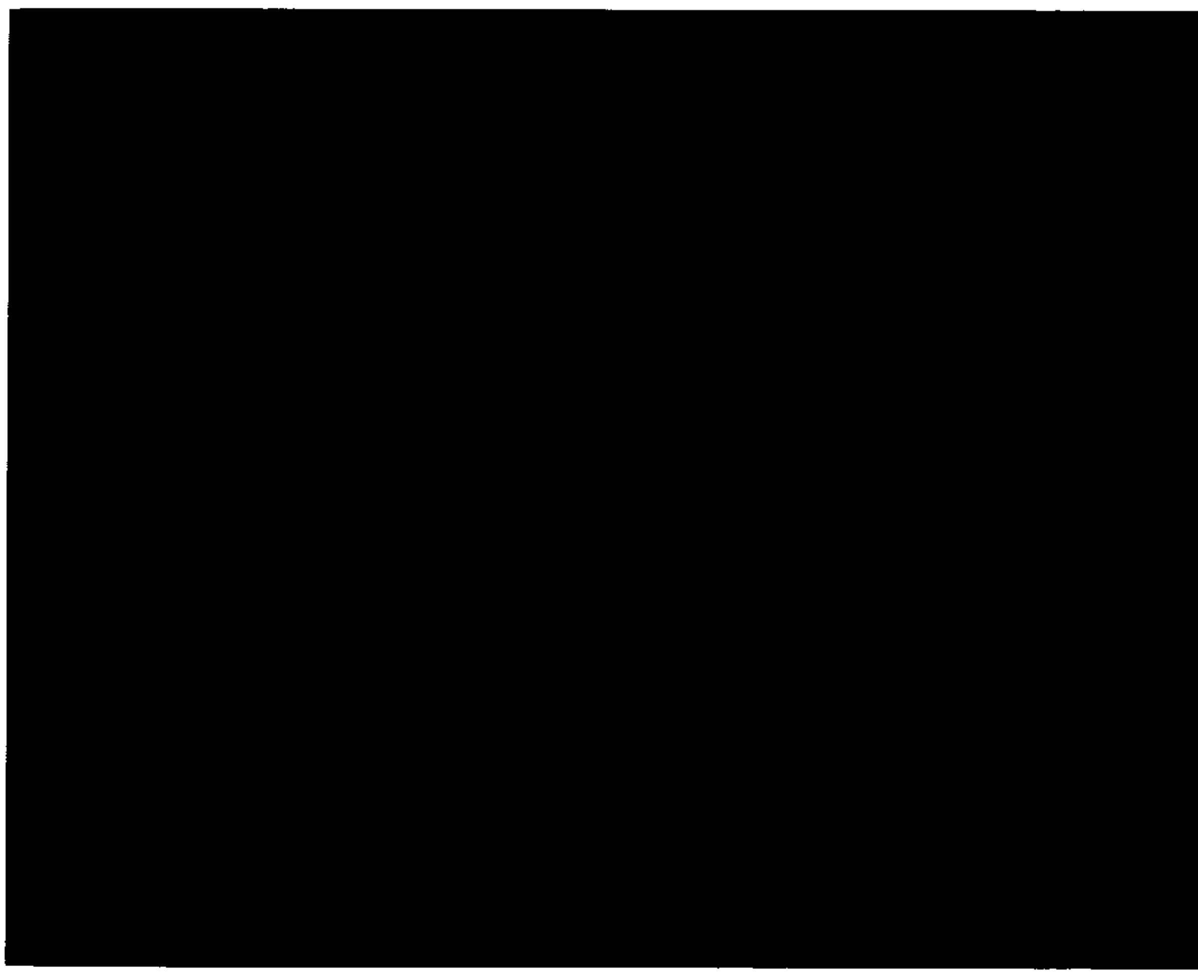
Vol. 3, No. 4
October - December 2007

**An official publication of
the Information Resources
Management Association**

INTERNATIONAL JOURNAL OF KNOWLEDGE MANAGEMENT



IGI PUBLISHING
Publisher of IT books, journals and cases since 1988
www.igi-pub.com

The *International Journal of Knowledge Management* (ISSN 1548-0666; eISSN 1548-0658) is published quarterly by IGI Global, 701 E. Chocolate Avenue, Suite 200, Hershey, PA 17033-1240, USA. For pricing, please visit www.igi-pub.com/journals. Sample copies may be requested (based on availability). **Advertising information:** IGI Global, 701 E. Chocolate Avenue, Suite 200, Hershey, PA 17033-1240, USA, 717/533-8845, <www.igi-pub.com>. **Postmaster:** Send all address changes to above address.

IGI PUBLISHING

Hershey • New York

Order online at www.igi-pub or call 717-533-8845 x10 –
Mon-Fri 8:30 am - 5:00 pm (est) or fax 24 hours a day 717-533-8661

International Journal of Knowledge Management

October-December 2007, Vol. 3, No. 4

Table of Contents

Editorial Preface

Research Papers

ledge
ction
urnal
ation

Developing Knowledge Management Systems from a Knowledge-Based and Multi-Agent Approach

Aurora Vizcaíno, Alarcos Research Group, University of Castilla-la Mancha, Spain

Juan Pablo Soto, Alarcos Research Group, University of Castilla-la Mancha, Spain

Javier Portillo-Rodríguez, Alarcos Research Group, University of Castilla-la Mancha, Spain

Mario Piattini, Alarcos Research Group, University of Castilla-la Mancha, Spain

ABSTRACT

Developing knowledge management systems is a complicated task since it is necessary to take into account how the knowledge is generated, how it can be distributed in order to reuse it, and other aspects related to the knowledge flows. On the other hand, many technical aspects should also be considered such as what knowledge representation or retrieval technique is going to be used. To find a balance between both aspects is important if we want to develop a successful system. However, developers often focus on technical aspects, giving less importance to knowledge issues. In order to avoid this, we have developed a model to help computer science engineers to develop these kinds of systems. In our proposal we first define a knowledge life cycle model that, according to literature and our experience, ponders all the stages that a knowledge management system should give support to. Later, we describe the technology (software agents) that we recommend to support the activities of each stage. The article explains why we consider that software agents are suitable for this end and how they can work in order to reach their goals. Moreover, a prototype that uses these agents is also described.

Keywords: *knowledge-based software; knowledge utilization*

INTRODUCTION

In the last decades, knowledge management (KM) has captured enterprises' attention as one of the most promising ways to reach success in this information era (Malone, 2002). A shorter life cycle of products, globalization, and strategic alliances between companies demand a deeper and more systematic organizational knowledge management. Consequently, one way to assess an organization's performance is to determine how well it manages its critical knowledge.

In order to assist organizations to manage their knowledge, systems have been designed. These are called knowledge management systems (KMS), defined by Alavi and Leidner (2001) as IT-based systems developed to support/enhance the processes of knowledge creation, storage/retrieval, transfer, and application.

However, developing KMS is a difficult task; since knowledge *per se* is intensively domain dependent whereas KMS often are context specific applications. Thus, reusability is a complex issue. On the other hand, the lack of sophisticated methodologies or theories for the extraction of reusable knowledge and reusable knowledge patterns has proven to be extremely costly, time consuming, and error prone (Gkotsis, Evangelou, Karacapilidis & Tzagarakis, 2006). Moreover, there are several approaches towards KMS developing. For instance, the process/task based approach focuses on the use of knowledge by participants in a project, or the infrastructure/generic system based approach focuses on building a base system to capture and distribute knowledge for use throughout the organization (Jennex, 2005). On the other hand, before developing this kind of system it is advisable to study and understand how the transfer of knowledge is carried out by people in real life. However, when developing KMS, developers often focus on the technology without taking into account the fundamental knowledge problems that KMS are likely to support (Hahn & Subramani, 2000).

Different techniques have been used to implement KMS. One of them, which is proving to be quite useful, is that of intelligent agents (van Elst, Dignum & Abecker, 2003). Software agent technology can monitor and coordinate events or meetings and disseminate information (Wooldridge & Jennings, 1995). Furthermore, agents are proactive in the sense that they can take the initiative and achieve their own goals. The autonomous behavior of the agents is critical to the goal of this research since it can reduce the amount of work that employees have to perform when using a KM system. Another important issue is that agents can learn from their own experience. Consequently, agent systems are expected to become more efficient with time since the agents learn from their previous mistakes and successes (Maes, 1994).

Because of these advantages, different agent-based architectures have been proposed to support activities related to KM (Gandon, 2000). Some architectures have even been designed to help in the development of KMS. However, most of them focus 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 article we propose a generic model for developing KMS. Therefore, in the next section we describe the model and the software agents that we propose to support it. In the following section, we explain how the agents are structured and how they have been modeled using the INGENIAS methodology. Later, the next section describes a prototype that we are implementing by using the agents proposed in the model. The following section summarizes related works carried out with agents. Finally, conclusions and future work are outlined in the last section.

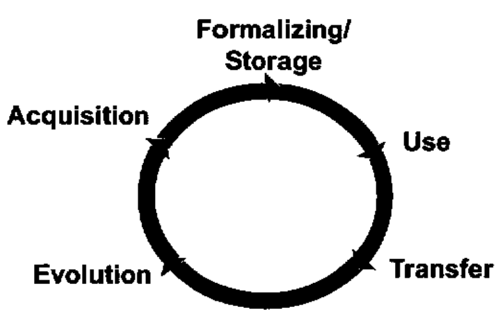
A MULTI-AGENT MODEL TO DEVELOP KNOWLEDGE MANAGEMENT SYSTEMS

A successful KMS should perform the functions of knowledge creation, storage/retrieval, transfer, and application (Jennex & Olfman, 2006). Taking this fact into account and after reviewing several knowledge life cycles and models (see Table 1) and seeing what stages most authors considered, we decided to define a knowledge life cycle that indicates what process a KMS should support (see Figure 1). This is a focus different to the previous one based on describing the knowledge cycle in human beings and/or in companies.

Table 1. Knowledge life cycle

Model	Stage1	Stage2	Stage3	Stage4	Stage5	Stage6	Stage7
Nonaka and Takeuchi (Nonaka & Takeychi, 1995)	Socialization	Externalization	Combination	Internalization			
Wiig (Wiig, 1997)	Creation	Storing/ gathering	Use	Leverage	Sharing		
Davenport and Prusak (Davenport & Prusak, 1998)	Generation	Codify/ Coordinate	Transfer	Roles and Skills			
Tiwana (Tiwana, 2000)	Acquire	Sharing	Use				
Alavi and Leidner (Alavi & Leidner, 2001)	Creation	Storage/ Retrieval	Transfer	Application			
Rus and Lindvall (Rus & Lindvall, 2002)	Creation/ Acquisition	Organization/ Storage	Distribution	Application			
Nissen (Davenport, 1998)	Creation	Organization	Formalize	Distribute	Application	Evolve	
Ward and Aurum (Ward & Aurum, 2004)	Creation	Distribution	Organization	Adaptation	Identification	distribution	Application
Dickinson (Dickinson, 2000)	Identification	Acquisition	Development	Distribution	Use	Preservation	

Figure 1. Knowledge life cycle model proposed



The stages of our proposal are acquisition, storage, use, transfer, and evaluation. The first three stages are considered in most knowledge life cycles (see Table 1). We have added transfer (also considered in several cycles) and evolution. Transfer is added because a KMS should disseminate knowledge to those people that can need it. Evolution is added because knowledge should always be updated otherwise it would not be used.

In the following paragraphs each stage of the model is described. At the same time and with the goal of illustrating that it is possible to support each stage by using current technology, we are going to explain how a software agent could be implemented for a KMS.

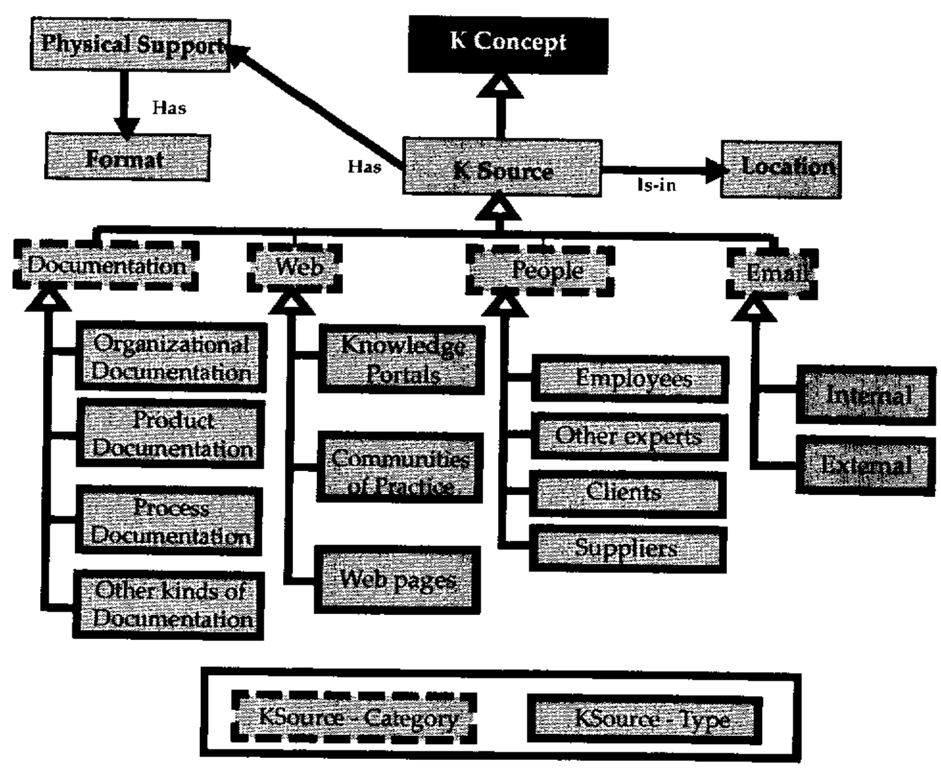
a. *Knowledge acquisition* is a key component of a KMS architecture. This stage includes the elicitation, collection, and analysis of knowledge (Rhem, 2006). During this process, it is vital to determine where in the organization the knowledge exists and how to capture it. The definition of the knowledge to be acquired can be assisted by classifying types of knowledge and knowledge sources (Dickinson, 2000). To support this stage we propose to use an agent called a *Captor Agent*. 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 the system or groupware tools (e-mail, consulted Web pages, chats, etc.). In order to accomplish this, the Captor Agent can use different techniques to acquire knowledge since there are several tools and techniques that consolidate and transform corporate data into information (Houari & Homayoun Far, 2004). They contain:

- Front-end system (i.e., decision support system [DSS], executive information system [EIS], and online analytical processing [OLAP]).
- Back-end system: data warehouse, data mart, and data mining (Giannella, Bhargava et al., 2004).

Agents can also apply classical techniques used by experts to acquire knowledge such as structured interviews, questionnaires, goal trees, decision networks, repertory grids, or conceptual maps (Rhem, 2006). More sophisticated techniques such as WebParser (Camacho, Aler & Cuadrado, 2004) to obtain information from the Web, document classification (Novak, Wurst, Fleschmann & Strauss, 2003), mailing list management (Moreale & Watt, 2003), or data mining and neuronal nets can be also used.

Once the knowledge has been obtained, the Captor Agent can classify it by using ontologies according to its type and the knowledge source from it was obtained (see Figure 2). This ontology is based on Rodriguez's ontologies for

Figure 2. Knowledge source ontology



representing knowledge topics and knowledge sources (Rodríguez, Martínez, Favela, Vizcaíno & Piattini, 2004).

The ontology has four knowledge source categories: Documentation, which can be subdivided into documentation related to the organization's philosophy, documentation which describes the product/s which the company works with, documentation that describes the process that the company carries out, and other types of documentation that an organization has but that cannot be classified into any of the previous subgroups. Another important source where the Captor Agent finds information is the Web, which can also be divided into other subcategories such as portals, communities of practice, and so forth. The main knowledge source in a company is, without any doubt, people. Depending on the type of company, people may be classified as clients, employees, and so forth. The last knowledge source that the Captor Agent can use is e-mail that can be classified as internal mail (mail sent between employees), and external mail (e-mails sent to other people outside the organization).

One advantage of this approach is that the Captor Agent can work in any domain since by changing these ontologies the Captor knows what key knowledge should be found and where it might be.

b. *Knowledge formalizing/storing* is the stage that groups all the activities that focus on organizing, structuring, representing, and codifying the knowledge with the purpose of facilitating its use (Davenport & Prusak, 1998). To help carry out these tasks we propose a *Constructor Agent*. This 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 is 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. In this way, the combination process of the SECI (Nonaka, 1994) model is carried out, producing new knowledge resulting in the merging of explicit knowledge plus new explicit knowledge.

Different techniques exist to store knowledge and frequently the technique used is narrowly related to the retrieval method used. Therefore, if a case-based reasoning is going to be used, the knowledge will be stored as "cases." Other techniques are knowledge objects, frames, predicate logic, or fuzzy logic. In the case of using ontologies to classify the knowledge, methodologies to represent the knowledge can be used. Examples of these methodologies are Ontolingua (Gruber, 1993) or Representation Formalism for Software Engineering Ontologies (REFSENO) (Tautz & Von Wangenheim, 1998).

c. Knowledge Use is one of the main stages, since knowledge is helpful when it is used and/or reused. The main enemy of knowledge reuse is ignorance. Employers often complain because employees do not consult knowledge sources and do not take advantage of the knowledge capital that the company has. KMS should offer the possibility of searching for information; they can even give recommendations or suggestions with the goal of helping users to perform their tasks by reusing lessons already learnt, as well as previous experiences. In our model the agent in charge of this activity is the Searcher Agent, which searches in the knowledge base for the needed knowledge. Different techniques are currently used to search for knowledge. Many of them are based on the use of the position and frequency of keywords

(Mohammadian & Jentzsch, 2004) or on information retrieval techniques (Frakes & Baeza-Yates, 1992; Liang & Huang, 2000). Other authors such as Sung Kim (2004) mix several techniques, data mining, and case-based reasoning to develop a recommender system.

d. *Knowledge Transfer* is the most investigated stage in knowledge management (Peachey, Hall & Cegielski, 2005). This stage is in charge of transferring tacit and explicit knowledge. Tacit knowledge can be transferred if it has been previously stored in shared means, for example, repositories, organizational memories, databases, and so forth. The transfer stage can be carried out by using mechanisms to inform people about the new knowledge that has been added. For this stage we propose a *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 (Wenger, 1998). An appropriate knowledge management linked to communities of practice helps to improve the organization's performance (Lesser & Storck, 2001). 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 with a new type of agent called the *personal agent* which is in charge of determining the users' profiles (it will be described in next section). Comparing this stage with the SECI model we can say that 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.

e. *Knowledge Evolution*. This stage is responsible for monitoring the knowledge that evolves daily. To carry out this activity we propose a *Maintenance Agent*. The main purpose of this agent is to keep the knowledge stored in the knowledge base updated. Therefore, information that is not often used is considered by the Maintenance Agent as information to be possibly eliminated.

MULTI-AGENTS AGENCIES

Once the model and the agents that we propose to give support to the different stages have been described, we are going to explain how the agents are structured into two agencies. Therefore, we group all the agents closely in charge of managing knowledge and supporting the different stages of the model proposed in one agency. Auxiliary agents are in another agency (see Figure 3).

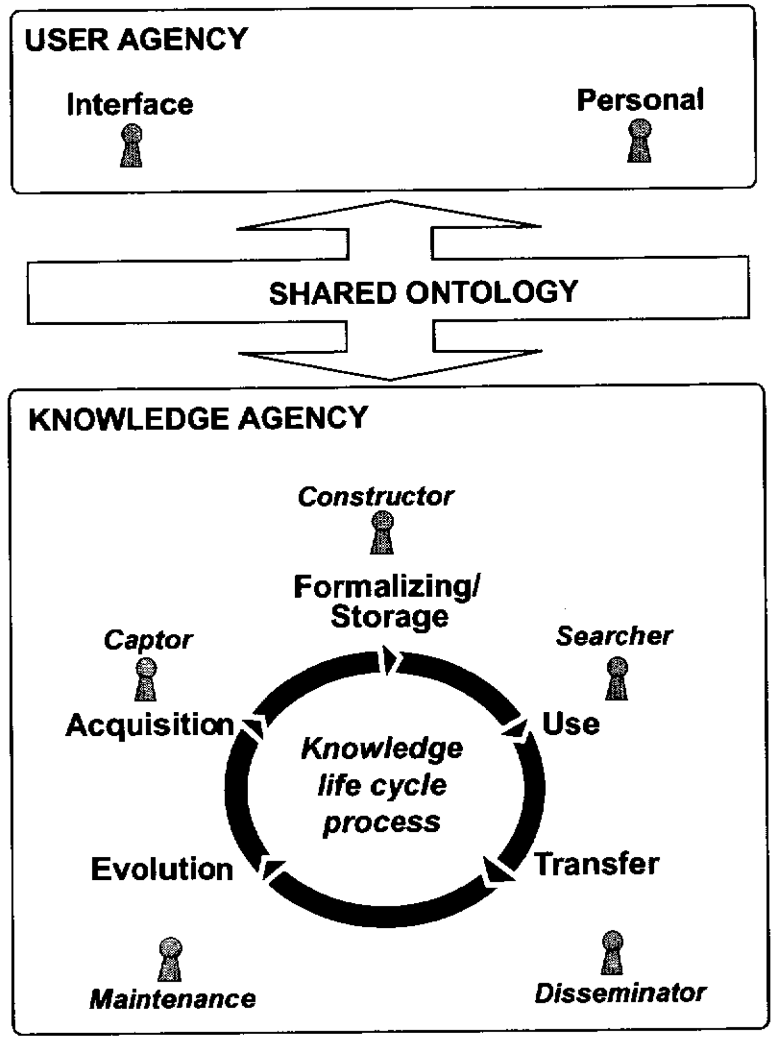
Therefore, the *Knowledge Agency* is in charge of giving support to the KM process. It consists of the Constructor Agent, the Captor Agent, the Searcher Agent, the Disseminator Agent, and the Maintenance Agent.

On the other hand, the *User Agency* is formed of the Personal Agent and the Interface Agent. The Personal Agent monitors users' tasks to obtain their preferences and needs. In order to implement the Personal Agent, user modeling techniques can be used. User modeling implies obtaining certain knowledge about the user. This knowledge describes what the user "likes" or what the user "knows" (Chin, 1986).

The *Interface Agent* is the mediator between the users and the agents. Thus, when an agent wants to communicate a message to the user, the agent sends the message to the Interface Agent which shows it to the user.

Another component is the *Shared Ontology* which provides a conceptualization of the knowledge domain. The Shared Ontology is

Figure 3. Agents distribution



used for the consistent communication of the agencies.

In order to carry out the analysis and design of the agents involved we have followed a methodology called INGENIAS (Pavón & Gómez-Sanz, 2003) which provides metamodels to define multi-agent systems, and support tools to generate them. Using metamodels facilitates the development of systems enormously, since they are oriented towards visual representations of concrete aspects of the system.

Below, we are going to show the different agent meta-model diagrams which describe the roles and tasks of each agent.

Figure 4 shows that the goal of the Captor Agent is to obtain information that should be stored. Its role is "filter" since it must decide what information should be transformed into knowledge, the purpose being to use this in future projects. In the following lines, we describe each of the tasks carried out by this agent.

- **IdentifyIS:** This task consists of identifying available knowledge sources in the system.
- **CaptureInfo:** The agent must also capture information.
- **SendToConstructor:** Once the suitability of storing the information has been analyzed, the Captor sends it to the Constructor Agent (described in Figure 5) whose roles are sculptor and treasurer since it is in charge of giving an appropriate electronic format to the information (sculptor) and of storing it in the knowledge base (treasurer).

The tasks developed by Constructor Agent are:

- **CompareInfo:** The agent is in charge of comparing the new information with the previously stored knowledge.
- **CombineInfo:** The agent is also in charge of combining the new information with the previously stored knowledge.
- **ClassifyInformation:** Another task is to classify the information received by the Captor Agent (for instance: models, structures, files, diagrams, etc.).
- **SendToDisseminator:** This is a critical task which consists of sending knowledge to the Disseminator Agent.
- **SaveKnowledge:** One of the most important tasks is to store the new knowledge into the knowledge base.

The Disseminator Agent, whose role is PostOfficeEmplee, as it behaves the "postman" of the architecture, (see Figure 6) is composed of the next tasks:

- **SaveInfoTemp:** The Disseminator Agent stores temporally the new knowledge received by the Constructor Agent.
- **EvaluateProfiles:** Once identified one user profile, the Disseminator Agent evaluates it in order to determine user's needs.
- **LookForActivePersonalAgents:** Personal Agents can be distributed into dif-

Figure 4. Captor agent diagram

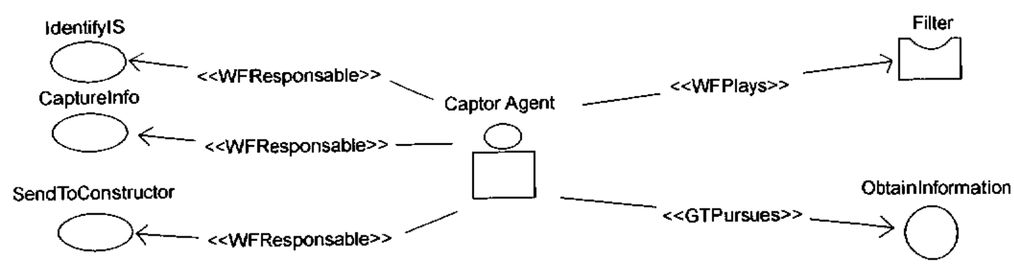


Figure 5. Constructor agent diagram

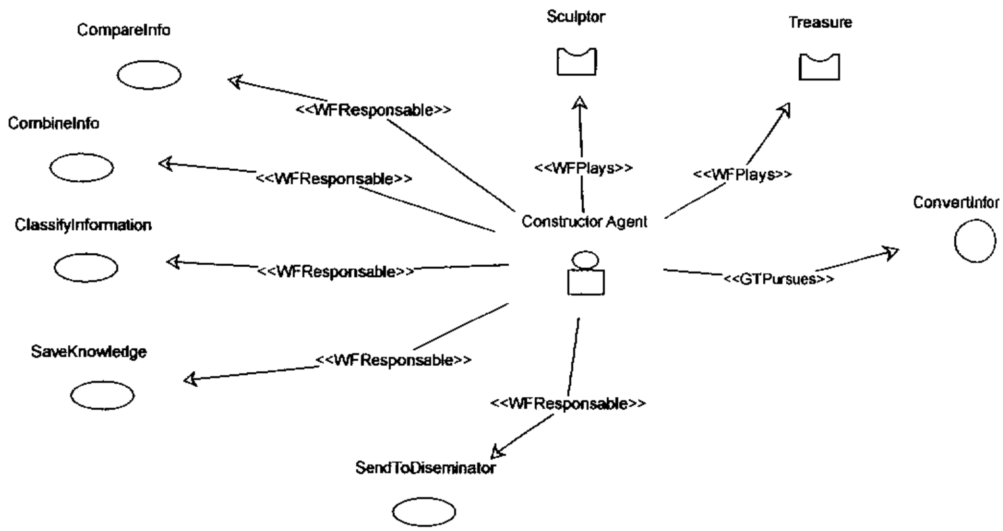
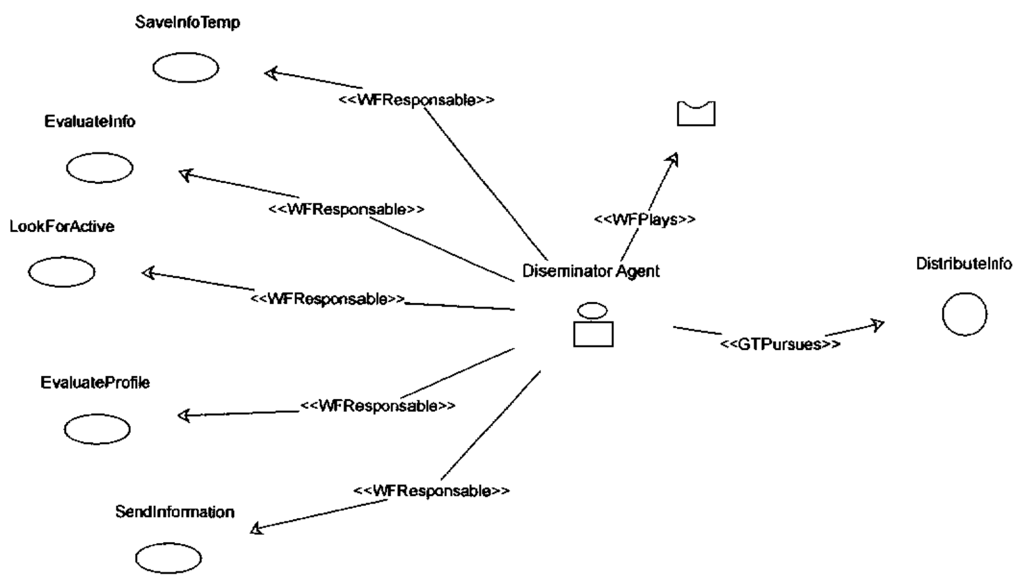


Figure 6. Disseminator agent diagram



ferent nodes, so it must identify all active Personal Agents available in the system.

- **SendInformation:** This is a critical task which consists of distributing the information to those people that can need it (really, the information is sent to their interface agents).
- **EvaluateInfo:** This task is focused on evaluating received information to be able to relate it with different user's profiles.

Another agent that supports 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 Agent diagram (Figure 7) is composed of the next tasks:

- **LookForInfo:** This agent is in charge of searching the information required by the users.
- **ClassifyInfo:** This agent also classifies the information found in the knowledge base.
- **SendInfoToI:** Finally, the agent sends the knowledge found in the knowledgebase to an Interface Agent.

The last type of agent of the Knowledge Agency is the Maintenance Agent (Figure 8). The main purpose of this agent is to keep the knowledge stored in the knowledge base updated. Therefore, its task dealt mainly with deleting obsolete information.

Now, the two types of agents of the User agency are described. Figure 9 shows the Personal Agent diagram whose role is called "spy" since the agent must monitor users' activities in order to obtain their profiles. Therefore, its goals are monitoring users' tasks and recommending information.

In order to attain these goals it should carry out the following tasks:

- **Modeling the users' profiles:** By observing the users' preferences, activities, information consulted, and so forth.
- **CreateManageLocalKnowledgeBase:** Creating and managing a "local knowledge

Figure 7. Searcher agent diagram

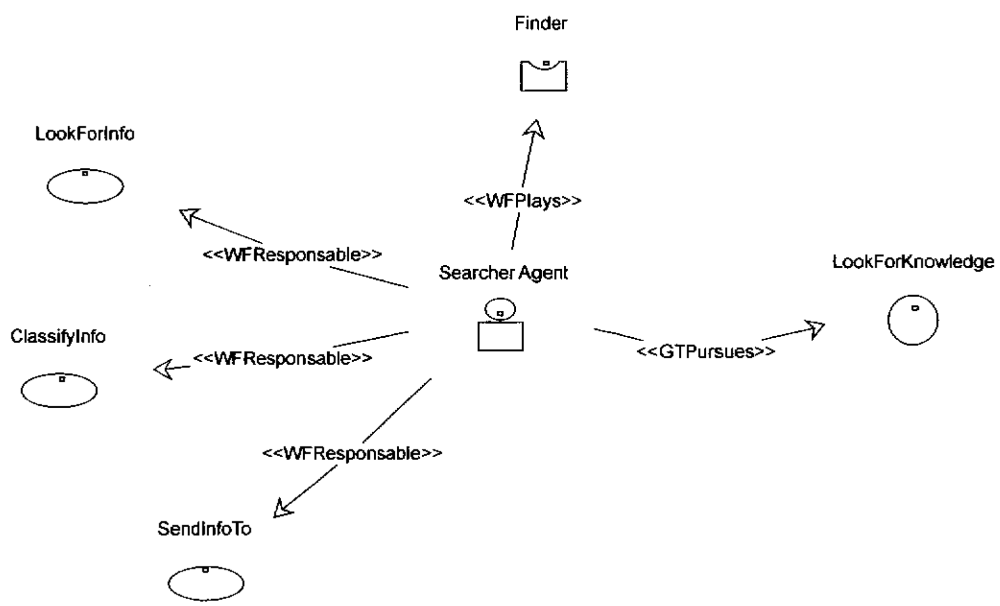


Figure 8. Maintenance agent diagram

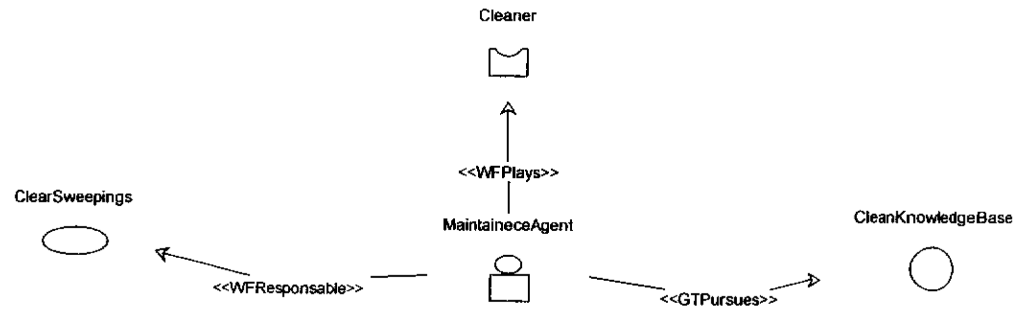


Figure 9. Personal agent diagram

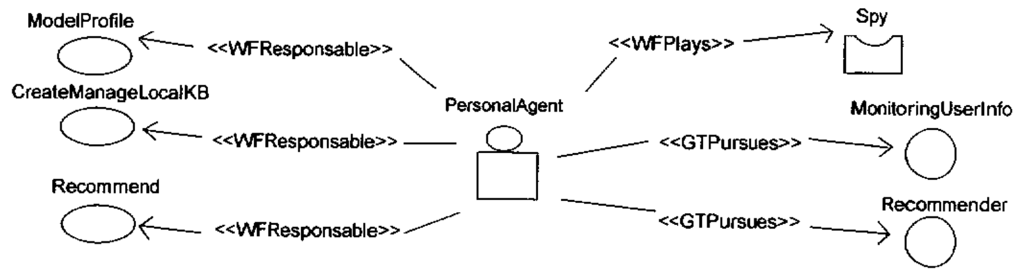
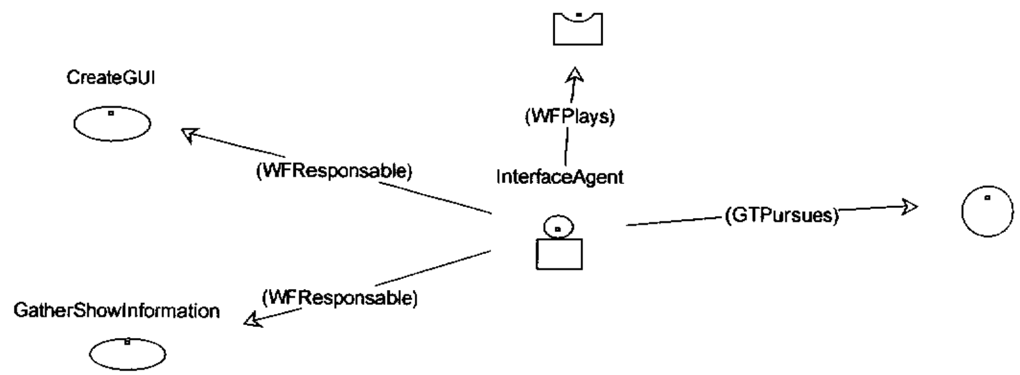


Figure 10. Interface agent diagram



base" where the relevant information for the user can be stored.

- **Recommending knowledge or knowledge sources:** This agent tries to guess what knowledge would be relevant for the user. To accomplish this, this agent communicates with the Searcher Agent and with the Interface agent.

On the other hand, the Interface Agent is an intermediary between the users and the rest of agents. Figure 10 shows that its main tasks are creating GUI and showing information to the users

These tasks are defined in order to attain the goal of showing important information to the user, named in the diagram ShowInformation, so we have to create an user interface and put the received information from others agents in a nice way to the user.

A PROTOTYPE SYSTEM

In order to test our model we are developing a KMS to be used in software maintenance companies. So far, the prototype recommends what information sources maintainers should consult to solve a particular problem. Before constructing the prototype, the knowledge flows that take place in software maintenance companies were studied (Rodríguez, Martínez, Vizcaíno, Favela & Piattini, 2005). To illustrate how the prototype works let us describe a scenario.

Scenario

A software maintenance engineer selects the project to be work on. Then, the employee starts to work on an activity (for instance a maintenance request). At the same time, the Personal Agent is monitoring the engineer's movements and is logging in what project and activity the engineer is working on. So, the Personal Agent sends the Searcher Agent a message asking for knowledge related to the activity that the employee is carrying out. Depending on the activity, the Searcher Agent can use two retrieval techniques, position and frequency of keywords in the case of needing to give information about a topic, or case-based reasoning in the case of having to propose a solution to a problem. When the Searcher Agent finds suitable information, the agent sends it to the Interface Agent, which is in charge of communicating to the employee that certain information exists which can be useful for the employee's work. The employee will decide if to consult this information. Figure 11 despites the diagram of this part.

Once the employee finishes the work, the Captor Agent checks whether a new case can be constructed (in case the employee had found a solution to a problem) or whether a new knowledge source has been used. In both cases the Captor sends the new knowledge to the Constructor Agent which is in charge of storing this in the knowledge base or adding new concepts to the knowledge source ontology according to the circumstance that have taken place.

The collaboration between the Captor and the Constructor Agent is depicts in Figure 12, which is an interaction model diagram that the INGENIAS methodology utilizes. These diagrams are very useful to see, at first glance, as agents interact.

Some Implementation Aspects

The platform that we are using to develop the architecture is java agent development framework (JADE) since it is FIPA compliant and is currently one of the most widely used. Moreover, JADE has been successfully used in the development of other systems in the domain of knowledge management (Bergenti, Poggi & Rimassa, 2000; Gandon, 2000).

RELATED WORK

Traditional KM systems have received certain criticism, since employees are often overloaded with extra work as they have to introduce information into the KMS and worry about updating this information. One proposal to avoid this extra burden was to add software agents to perform this task in place of the employees. Later, intelligent agent technology was also applied to other different activities, bringing several benefits to the knowledge management process.

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 knowledge management tasks or applications in which an agent can assist is very wide, for instance:

- To manage organizational memory, an example being the CoMMA project, (Gandon, 2000) (Corporate Memory Man-

Figure 11. Scenario diagram

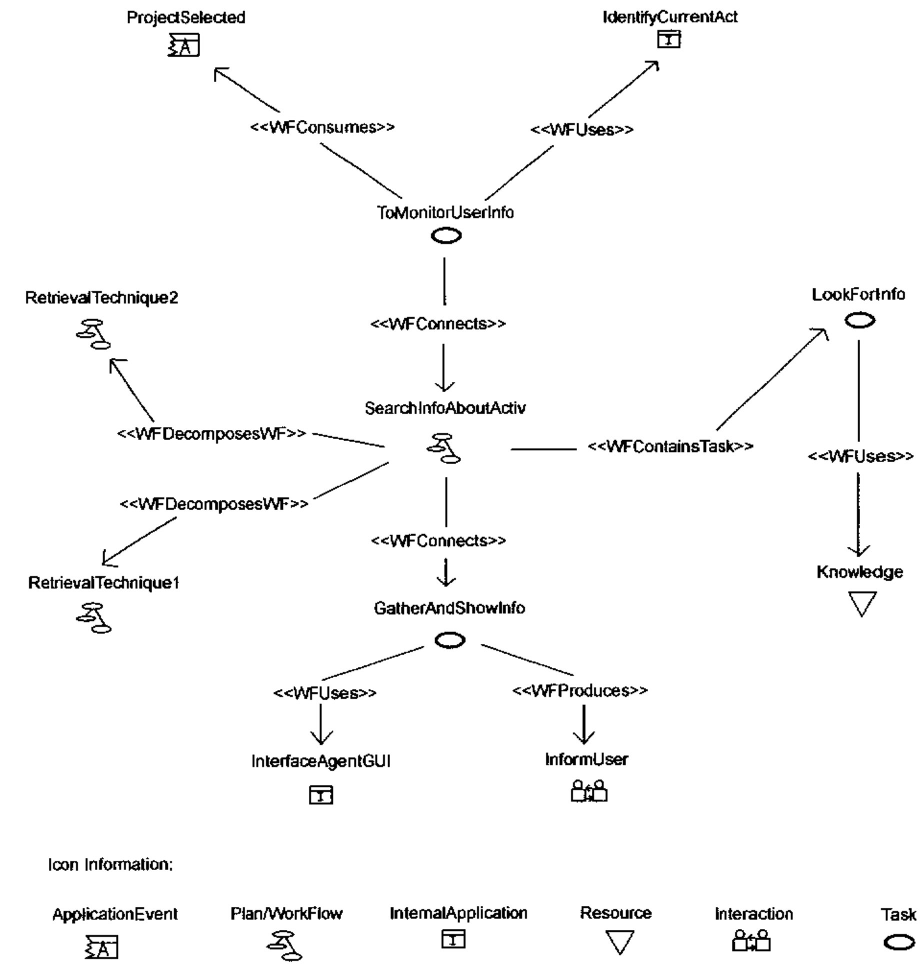
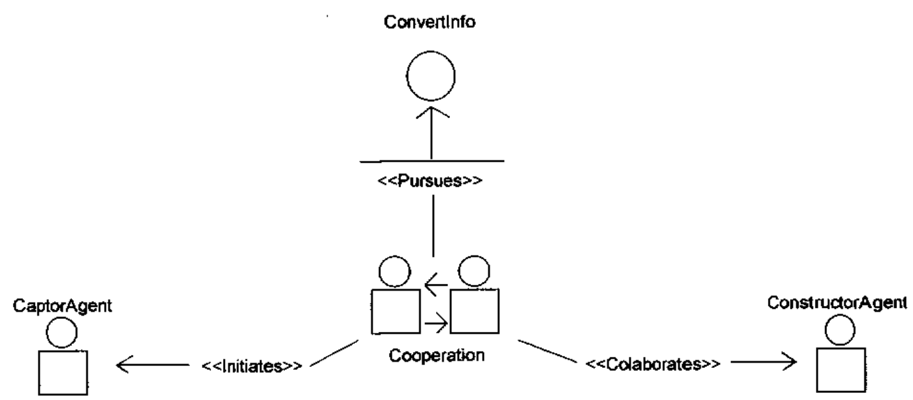


Figure 12. Cooperation between captor and constructor agent



agement through Agents), which combines emergent technologies, allowing users to exploit an organizational memory.

- To support cooperative activities. For instance Wang, Reidar, and Chunnian (1999) propose a multi-agent architecture to provide support to cooperative activities.
- To recommend. For instance Sung Kim (2004) describes a system to customize recommendations.
- To find experts. Some systems are used to help people find experts who can assist them in their daily work.
- To share knowledge. For instance Mercer and Greenberg (2001) propose a multi-agent system for knowledge sharing in a system designed to advise good programming practice.
- To manage mailing lists, or document classification (Moreale & Watt, 2003).

These and other existing systems were often developed without considering how knowledge flows and what stages may foster these flows. Because of this, they often support only one knowledge task, without taking into account that knowledge management implies giving support to different process and activities. On the other hand, KM systems often focus on the technology, without taking into account fundamental problems that these kinds of systems are likely to support (Hahn & Subramani, 2000).

CONCLUSION

The main contributions of this article are the design of knowledge cycle for developing KMS where the main functions that this kind of systems must support are described. Moreover, a multi-agent architecture is outlined to help KMS developers to implement these kinds of systems. The advantages of these contributions are:

- The model provides support to different activities: knowledge creation, storage/retrieval, transfer, and application. All are activities which, according to the authors who specialize in evaluating KMS, should support this kind of system.
- The architecture is based on a KM life cycle that we have proposed for this end. Therefore, we try to avoid the lack of other architectures that are focused on the technology and forget the knowledge aspects.
- The architecture makes use of intelligent agents. This is a technique that have proved to be very convenient in knowledge management activities since it avoids one of the problems of some KMS such as overloading the employees with extra work instead of helping them during their daily work. Agents can carry out many tasks on behalf of users. Moreover, they act when they consider that it is necessary to do so without needing users' instructions. Another advantage of using agents is that they can collaborate with other agents already implemented to carry out concrete knowledge tasks; for instance, obtaining information from the Internet or from e-mail. Thus, the development of KMS would be easier since only the basic agents of our model would have to be implemented and these could collaborate with other agents that have already been tested.

On the other hand, we are modeling the agents in a systematic way by using INGENIAS methodology whose metamodels help future developers to understand how the different agents work.

As future work we aim to compare the implementation of a KMS based on our proposal with developments using other architectures. Without any doubt this evaluation will help us to improve our proposal. On the other hand, we are also working on extending the model documentation with a more wide and detailed description of the possible techniques that could be used to implement each type of agent according to the main needs that organizations usually demand.

From a technological point of view, we are also studying JADEx in order to see how easy it would be to migrate to this new platform.

ACKNOWLEDGMENT

An earlier version of this article was presented at the 40th Hawaii International Conference on System Sciences (HICSS-2007), 03-06 January 2007, Waikoloa, Big Island, Hawaii.

This work is partially supported by the ENIGMAS (PIB-05-058), and MECENAS (PBI06-0024) project, Junta de Comunidades de Castilla-La Mancha, Consejería de Educación y Ciencia, both in Spain. This is also supported by the ESFINGE project (TIN2006-15175-C05-05) Ministerio de Educación y Ciencia (Dirección General de Investigación)/Fondos Europeos de Desarrollo Regional (FEDER) in Spain.

REFERENCES

Alavi, M., & Leidner, D. E. (2001). Knowledge management and knowledge management systems: Conceptual foundations and research issues. *MIS Quarterly, 25*(1), 107-136.

Bergenti, F., Poggi, A., & Rimassa, G. (2000). *Agent architecture and interaction protocols for corporate memory management systems.* Paper presented at the ECAI'2000, Workshop on Knowledge Management and Organizational Memories.

Camacho, D., Aler, R., & Cuadrado, J. (2004). Rule-based parsing for Web data extraction. In M. Mohammadian (Ed.), *Intelligent Agents for data mining and information retrieval* (pp. 65-87). Hershey, PA: Idea Group Publishing.

Chin, D. (1986). User modelling in UC: The UNIX consultant. *Human factors in computing systems.*

Davenport , P. (1998). *Working knowledge.*

Davenport, T. H., & Prusak, L. (1998). *Working knowledge: How organizations manage what they know.* Boston: Harvard Business School Press.

Dickinson, A. (2000). Enhancing knowledge management in enterprises (ENKE) IST project, IST-2000-29482. Retrieved April 27, 2007, from http://www.ist-enke.com

Frakes, W. B., & Baeza-Yates, R. (1992). *Information retrieval data structures and algorithms.* Englewood Cliffs, NJ: Prentice Hall.

Gandon, F. (2000). *A multi-agent architecture for distributed corporate memories.* Paper presented at the Third International Symposium From Agent Theory to Agent Implementation in European Meeting on Cybenertics and Systems Research.

Giannella, C., Bhargava, R. & K. H. (2004). *Multi-agent systems and distributed data mining.* Paper presented at the Cooperative Information Agents VIII: 8th International Workshop (CIA'04), Erfurt, Germany. Springer-Verlag.

Gkotsis, G., Evangelou, C., Karacapilidis, N., & Tzagarakis, M. (2006). *Building collaborative knowledge-based systems: A Web engineering approach.* Paper presented at the WWW/Internet, Iadis International Conference, Murcia.

Gruber, T. R. (1993). A translation approach to portable ontology specification. *Knowledge Acquisition, 5*(2), 199-220.

Hahn, J., & Subramani, M. (2000). *A framework of knowledge management systems: Issues and challenges for theory and practice.* Paper presented at the 21st International Conference in Informatin Systems (ICIS'00) (pp 302-312), Brisbane.

Houari, N., & Homayoun Far, B. (2004). *Application of intelligent agent technology for knowledge management integration.* Paper presented at the Third IEEE International Conference on Cognitive Informatics (ICCI'04).

Jennex, M. E. (2005). *The issue of system use in knowledge management systems.* Paper presented at the 38th Hawaii International Conference on System Sciences.

Jennex, M. E., & Olfman, L. (2006). A model of knowledge management success. *Journal of Knowledge Management, 2*(3), 51-68.

Lesser, E. L., & Storck, J. (2001). Communities of practice and organizational performance. *IBM Systems Journal, 40*(4), 831-841.

Liang, T. P., & Huang, J. S. (2000). A framework for applying intelligent agents to support electronic trading. *Decision Support Systems, 28*(4), 305-317.

Maes, P. (1994). Agents that reduce work and information overload. *Communications of the ACM, 37*(7), 31-40.

Malone, D. (2002). Knowledge management: A model for organizational learning. *International Journal of Accounting Information Systems, 3*, 111-123.

Mercer, S., & Greenberg, S. (2001). *A multi-agent architecture for knowledge sharing*. Paper presented at the Sixteenth European Meeting on Cybernetic and Systems Research, Viena.

Mohammadian, M., & Jentzsch, R. (2004). Computational intelligence techniques driven intelligent agents for Web data mining and information retrieval. In M. Mohammadian (Ed.), *Intelligent agents for data mining and information retrieval*. Hershey, PA: Idea Group Publishing.

Moreale, E., & Watt, S. (2003). An agent-based approach to mailing list knowledge management. *Agent-mediated knowledge management.*

Nonaka, I. (1994). A dynamic theory of organizational knowledge creation. *Organization Science, 5*(1), 14-37.

Nonaka, I., & Takeychi, H. (1995). *The knowledge-creation company: How Hapanese companies create the dynamics of innovation*. New York: Oxford University Press.

Novak, J., Wurst, M., Fleschmann, M., & Strauss, W. (2003). Discovering, visualizing and sharing knowledge through personalized learning knowledge maps. *Agent-mediated knowledge management*. Standford.

Pavón, J., & Gómez-Sanz, J. J. (2003). *Agent oriented software engineering with INGENIAS*. Paper presented at the Multi-Agent Systems and Applications III: CEEMAS 2003 (LNAI), Prage, Czech Republic. Springer.

Peachey, T., Hall, D., & Cegielski, C. (2005). Knowledge management and the leading information systems journals: An analysis of trends and gaps in published research. *International Journal of Knowledge Management, 1*(3), 55-69.

Rhem, A. J. (2006). *UML for developing knowledge management systems*. New York: Auerbach Publications.

Rodríguez, O., Martínez, A. I., Favela, J., Vizcaíno, A., & Piattini, M. (2004). *Understanding and supporting knowledge flows in a community of software developers*. Paper presented at the 10th International Workshop on Groupware (CRIWG 2004) LNCS 3198, San Carlos, Costa Rica. Springer.

Rodríguez, O. M., Martínez, A. I., Vizcaíno, A., Favela, J., & Piattini, M. (2005). Identifying knowledge flows in communities of practice. In E. Coakes & S. A. Clarke (Eds.), *Encyclopedia of communities of practice in information and knowledge management*. Hershey, PA: Idea Group Publishing.

Rus, I., & Lindvall, M. (2002, May/June). Knowledge management in software engineering. *IEEE Software*, 26-38.

Sung Kim, J. (2004). Customized recommendation mechanism based on Web data mining and case-based reasoning. In M. Mohammadian (Ed.), *Intelligent agents for data mining and information retrieval.*.

Tautz, C., & Von Wangenheim, C. G. (1998, October 20). *REFSENO: A representation formalism for software engineering ontologies* (Fraunhofer IESE-Report N° 015.98/E, version 1.1).

Tiwana, A. (2000). *The knowledge management toolkit: Practical techniques for building a knowledge management system*. Prentice Hall.

van Elst, L., Dignum, V., & Abecker, A. (2003). Agent-mediated knowledge management. *Agent-mediated knowledge management*. Standford.

Wang, A., Reidar, C., & Chunnian, L. (1999). *A multi-agent architecture for cooperative software engineering*. Paper presented at the Eleventh International Conference on Software Engineering and Knowledge Engineering (SEKE'99), Germany.

Ward, J., & Aurum, A. (2004). *Knowledge management in software engineering: De-scribing the process*. Paper presented at the 15th Australian Software Engineering Conference (ASWEC 2004), Melbourne, Australia. EEE Computer Society Press.

Wenger, E. (1998). *Communities of practice: Learning, meaning, and identity*. Cambridge, U.K.: Cambridge University Press.

Wiig, K. M. (1997). Knowledge management: Where did it come from and where will it go? *Expert Systems with Applications, 13*, 1-14.

Wooldridge, M., & Jennings, N. R. (1995). Intelligent agents: Theory and practice. *The Knowledge Engineering Review, 10*(2), 115-152.

Aurora Vizcaíno is an associate professor at the Escuela Superior de Informática of the University of Castilla-La Mancha, Spain. She has obtained an MSc and an European PhD in computer science by the University of Castilla-La Mancha. Her PhD work was based on the use of a simulated student in collaborative environments. Her research interests include collaborative learning, agents, simulated student and knowledge management. She has numerous publications in important international conferences and is part of the program committee of many of them.

Juan Pablo Soto is a computer science engineer by the University of Baja California (México). He is a PhD student and the topic of his thesis is based on the goal of improving knowledge management systems by using multi-agent technology. His research interests are knowledge management and multi-agent systems.

Javier Portillo is currently studying the last course of a degree in computer science and he is also a technician working at the Alarcos Research Group, (University of Castilla-La Mancha) Spain. His research interests include Agent Technology and Knowledge Management.

Mario Piattini has an MSc and PhD in computer science by the Polytechnic University of Madrid. Certified information system auditor by ISACA (Information System Audit and Control Association). Full professor at the Escuela Superior de Informática of the Castilla-La Mancha University. He is also the author of several books and papers on databases, software engineering and information systems. He leads the ALARCOS research group of the Department of Computer Science at the University of Castilla-La Mancha, in Ciudad Real, Spain. His research interests are: advanced database design, database quality, software metrics, object oriented metrics, software maintenance.