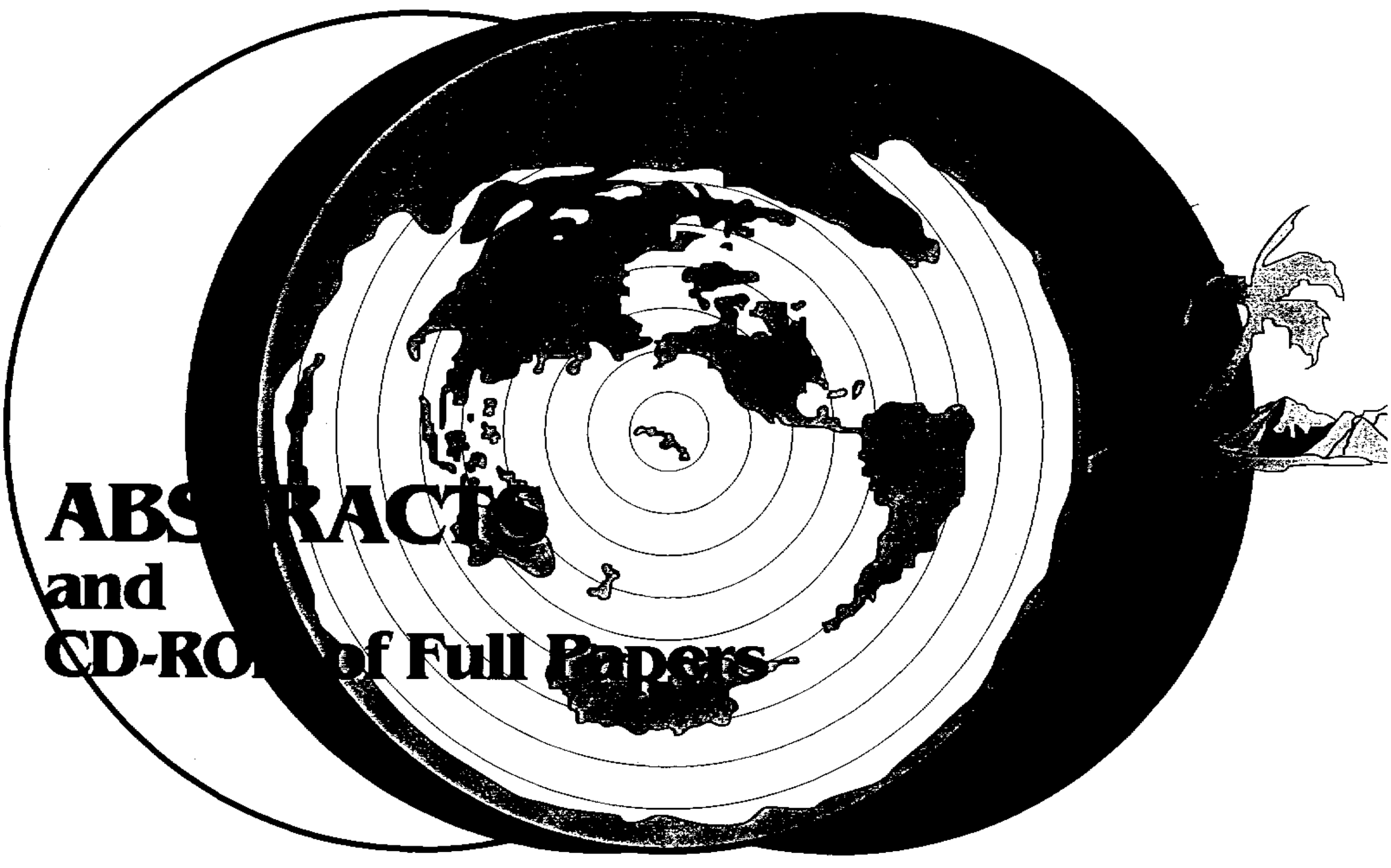
Proceedings of the Fortieth Annual Hawaii International Conference on

System Sciences

Waikoloa, Big Island, Hawaii **2007** **3-6 January 2007**



ABSTRACTS and CD-ROM of Full Papers

UNIVERSITY OF HAWAI'I AT MĀNOA
SHIDLER
COLLEGE OF BUSINESS
• International Excellence •

Published by the IEEE Computer Society
10662 Los Vaqueros Circle
P.O. Box 3014
Los Alamitos, CA 90720-1314

IEEE Computer Society Order Number P2755
ISSN 1530-1605
ISBN 0-7695-2755-8

ISBN 0-7695-2755-8
90000
9 780769 527550

Proceedings of the 40th Annual Hawaii International Conference on System Sciences

Proceedings of the 40th Annual Hawaii International Conference on System Sciences

3-6 January 2007

Big Island, Hawaii

ABSTRACTS
and
CD-ROM of Full Papers

Edited by
Ralph H. Sprague, Jr.

IEEE COMPUTER SOCIETY

IEEE

Los Alamitos, California

Washington • Tokyo

CPS Conference Publishing Services

IEEE Computer Society Order Number P2755
ISBN-13: 978-0-7695-2755-0
ISBN-10: 0-7695-2755-8
ISSN 1530-1605

Additional copies may be ordered from:

IEEE Computer Society
Customer Service Center
10662 Los Vaqueros Circle
P.O. Box 3014
Los Alamitos, CA 90720-1314
Tel: + 1 800 272 6657
Fax: + 1 714 821 4641
http://computer.org/cspress
csbooks@computer.org

IEEE Service Center
445 Hoes Lane
P.O. Box 1331
Piscataway, NJ 08855-1331
Tel: + 1 732 981 0060
Fax: + 1 732 981 9667
http://shop.ieee.org/store/
customer-service@ieee.org

IEEE Computer Society
Asia/Pacific Office
Watanabe Bldg., 1-4-2
Minami-Aoyama
Minato-ku, Tokyo 107-0062
JAPAN
Tel: + 81 3 3408 3118
Fax: + 81 3 3408 3553
tokyo.ofc@computer.org

Individual paper REPRINTS may be ordered at: reprints@computer.org

HICSS Production Editor: Eileen Robichaud, CMP

Editorial production by Stephanie Kawada

Cover art production by Joe Daigle/Studio Productions

Printed in the United States of America by The Printing House

CD-ROM produced by InControl Productions, Inc.

Conference Publishing Services
http://www.computer.org/proceedings/

Preface

The Fortieth Hawaii International Conference on System Sciences (HICSS) was held at the Hilton Waikoloa Village on the Island of Hawai'i, on January 3-6, 2007. The proceedings consist of over 450 papers in ten major tracks. The topics are summarized on the Overview page which shows the tracks and minitracks. A new track has been added since last year—Electric Power Systems Restructuring: Engineering, Economics and Policy.

These Proceedings consist of papers that have not previously been published. They have undergone a detailed peer review process and were selected based on rigorous standards. At the conference, these papers were presented by the authors and discussed in highly interactive sessions. HICSS provides a unique forum for the interchange of ideas, advances, and applications among academicians and practitioners in the information, computing, and system sciences. HICSS is sponsored by Shidler College of Business at the University of Hawai'i.

The HICSS series of conferences is now in the 40th year. Very few conferences have been able to grow and develop as HICSS has over this period. The computer age is just over 50 years old, and HICSS has been an important event in the world of computer science and information technology applications during most of that time.

The conference continues to be one of the best working conferences in computer-related sciences, with a high level of interaction among the leading scientists, engineers, and professionals. Many of the papers and presentations at HICSS have become journal articles, have developed into monographs or books, or have appeared in special issues of journals of wide circulation such as *Computer*, *IEEE Software, Decision Support Systems*, and *Journal of Management Information Systems*. Some major topics such as Collaboration Systems, Model Management Systems and Negotiation Support Systems have been largely initiated, developed and documented by presentations and papers at HICSS. Sincere thanks go to all the authors, attendees, coordinators, chairpersons, advisory committees, and administrative staff who make the conference a success.

The HICSS proceedings have made an impressive contribution to the literature base with over 80,000 pages during the past 40 years. Since 1999, we have engaged in a process of expanding the Proceedings to include additional forms and media. The objective has been to capture the benefits of electronic media while preserving the benefits of printed Proceedings. The CD/ROM of full papers is included in a sleeve inside the back cover of the book of abstracts. Both are distributed at the Conference and sold afterwards by the IEEE Computer Society Press. You may peruse the abstracts to find papers of interest, then use the CD to read the full paper. The abstract book is arranged in alphabetical order by Track and Minitrack name. On the CD, you may use the excellent Electronic Guide or the search facility on the CD to find papers of interest.

Ralph H. Sprague, Jr.
HICSS-40 Conference Chair
E-mail Address: sprague@hawaii.edu

Proceedings Overview for HICSS-40

Conference Administration

Chairman
Ralph H. Sprague, Jr.
University of Hawai'i at Manoa

Conference Administrator
Sandra Laney
University of Hawai'i at Manoa

Track Administrator
Eileen Robichaud, CMP
Cactus Meetings & Technologies, Inc.

Track Chairs

Collaboration Systems and Technology
Jay F. Nunamaker, Jr., University of Arizona
Robert O. Briggs, University of Nebraska at Omaha and Delft University of Technology, Netherlands

Decision Technologies and Service Sciences
Daniel R. Dolk, Naval Postgraduate School

Digital Media: Content and Communication
Michael Shepherd, Dalhousie University, Canada

Electronic Government
Hans J. (Jochen) Scholl, University of Washington

Electric Power Systems Restructuring: Engineering, Economics and Policy
Robert J. Thomas, Cornell University

Information Technology in Health Care
William G. Chismar, University of Hawai'i

Internet and the Digital Economy
David R. King, JDA Software Group, Inc., Scottsdale, AZ
Alan Dennis, Indiana University

Knowledge Management Systems
Murray Jennex, San Diego State University
Dave Croasdell, University of Nevada, Reno

Organizational Systems and Technology
Hugh Watson, University of Georgia

Software Technology
Gul Agha, University of Illinois at Urbana-Champaign
Rick Kazman, University of Hawai'i at Manoa

Value Webs in the Digital Economy
Helmut Kcrmar, Technische Universitaet Muenchen, Germany
Kalle Lyytinen, Case Western Reserve University
Jan Marco Leimeister, Technische Universitaet Muenchen, Germany

Virtual Communities
Karine Barzilai-Nahon, University of Washington

Knowledge Management Systems Track

Murray Jennex, San Diego State University
Dave Croasdell, University of Nevada, Reno

Philosophy and Ethics in Knowledge Management
Jim Sheffield, University of Auckland, New Zealand
Robert Mason, University of Washington

Knowledge Management for Multi- and Crossdisciplinary Research Projects
Pieter J. Beers, Deflt University of Technology, Netherlands
Pieter W.G. Bots, Cemagref, Montpellier, France

Knowledge Flows: Knowledge Transfer, Sharing and Exchange in Organizations
K.D. Joshi, Washington State University
Mark E. Nissen, Naval Postgraduate School
Saonee Sarker, Washington State University

Knowledge Management/Organizational Memory Success and Performance Measurements
Murray E. Jennex, San Diego State University
Stefan Smolnik, University of Paderborn, Germany
Dave Croasdell, University of Nevada, Reno

Modeling Knowledge Intensive Processes
Balasubramaniam Ramesh, Georgia State University
Lan Cao, Old Dominion University

Social and Distributed Cognition in Knowledge Management Systems
Susan Gasson, Drexel University
Jim Waters, Drexel University

Technological Aspects of Knowledge Systems
Richard Orwig, Susquehanna University
Dianne Hall, Auburn University
Jim Courtney, University of Central Florida

Table of Contents

40th Annual Hawaii International Conference on System Sciences (HICSS-40)

Collaboration Systems and Technology Track

Co-Chairs: Jay F. Nunamaker, Jr. and Robert O. Briggs

Knowledge Management Systems Track
Co-Chairs: Murray Jennex and Dave Croasdell

Social and Distributed Cognition in Knowledge Management Systems
Co-Chairs: Susan Gasson and Jim Waters

Technological Aspects of Knowledge Systems
Co-Chairs: Richard Orwig, Dianne Hall, and Jim Courtney

The Co-evolution of Design and User Requirements in Knowledge Management Systems: The Case of Patent Management Systems

Tony Briggs, Bala Iyer, and Paul Carlile

Do the design requirements of a knowledge management system change over time? If so, how do these changes affect the users of the system? In this paper we explore the case of patent management systems (PAMS) to identify changes in information system design and user requirements over a 200 year period. Using a design science approach, we study 30 different implementations of PAMS across different design configurations and use scenarios. We find that while early forms of PAMS persist in updated system implementations, new design configurations coevolved with new user requirements. We conclude by suggesting that this co-evolutionary path is not unique to PAMS and suggest that a similar co-evolution is driving the development of new World Wide Web technologies.

A Multi-agent Model to Develop Knowledge Management Systems

Aurora Vizcaíno, Juan Pablo Soto, Javier Portillo, and Mario Piattini

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, firstly, we 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 paper 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.

Assessing User Acceptance of a Knowledge Management System in a Global Bank: Process Analysis and Concept Development

Cristof Bals, Stefan Smolnik, and Gerold Riempp

Many organizations pursue knowledge management (KM) initiatives, with different degrees of success. One key aspect of KM often neglected in practice is that it not only concerns technology. Technology merely provides the tools with which employees can leverage their knowledge in the context of their work. Thus, how employees perceive the technology and interact with it is assumed to play a major role in KM initiatives' success. This paper analyses patterns of user behavior and acceptance of knowledge management systems (KMS) to identify their relevance for a KM initiative's overall success. Using a combined single case study and literature review approach, we develop a model of user behavior and acceptance of KMS. By combining the user acceptance model with a model of context-specific influencing factors and the integrated KMS architecture by Riempp, we also present an integrated framework for approaching KM initiatives.

A Multi-agent Model to Develop Knowledge Management Systems

Aurora Vizcaíno, Juan Pablo Soto, Javier Portillo, Mario Piattini
Grupo Alarcos. Universidad de Castilla-la Mancha.
Ciudad Real. Spain
aurora.vizcaino@uclm.es, jpsoto@proyectos.inf-cr.uclm.es,
Javier.Portillo@alu.uclm.es, Mario.piattini@uclm.es

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, firstly, we 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 paper 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.

1. Introduction

Nowadays, knowledge is a key factor for an organization's competitive advantage; because of this the production environment and infrastructure play a diminishing role and intellectual capital and knowledge management a growing one [12]. 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 perform this task, systems have been designed to manage organizational knowledge. These are called Knowledge Management Systems (KMS), defined by Alavi and Leidner [1], as an IT-based system developed to support/enhance the processes of knowledge creation, storage/retrieval, transfer, and application.

Developing KMS is a difficult task; in fact, there are different approaches towards this. 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 [1]. 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 [11].

Different techniques have been used to implement KMS. One of them, which is proving to be quite useful, is that of intelligent agents [35]. Software agent technology can monitor and coordinate events or meetings and disseminate information [40]. 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 [18].

Because of these advantages different agent-based architectures have been proposed to support activities related to KM [8]. 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 paper we propose a generic model for developing KMS. Therefore, in section two we describe the model and the software agents that we propose to support it. In section three, we explain how the agents are structured and how they have been model by using INGENIAS methodology [26]. Later, section four describes a prototype that we are implementing by using the agents proposed in the model. Section five summarizes related works carried out with agents. Finally, conclusions and future work are outlined in section six.

2. A Multi-agent model to develop knowledge management systems

A successful KMS should perform the functions of knowledge creation, storage/retrieval, transfer and application [15]. Taking this fact into account and after reviewing several knowledge life cycles and models [1,5, 6, 22, 24, 31, 34, 37, 39] and seeing what stages most authors considered, we decided to define a knowledge life cycle that indicates what process a KMS should support (see Figure1). The stages of this cycle are acquisition, storage, use, application and evaluation. The first three stages are considered in most knowledge life cycles (see Appendix one). We have added use (also considered in many cycles) and evaluation since a KMS should foster users to utilize the knowledge that the company has (for instance, by recommending what information can be consulted or by proposing solutions to problems). Moreover, knowledge should always be updated otherwise it would not used. Because of this, the evolution stage has also been added to this cycle.

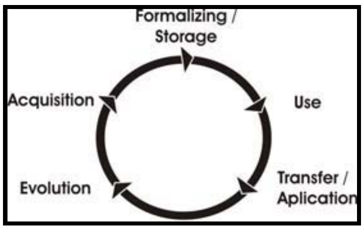


Figure 1. Knowledge Life Cycle Model Proposed

In the following paragraphs each stage of the model is described. At the same time, we are going to explain a software agent that could be implemented in a KMS to give support to each stage.

Knowledge acquisition is a key component of a KMS architecture. This stage includes the elicitation, collection, and analysis of knowledge [28]. 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 [6]. 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 (email, 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 [13]. They contain:

- Front-end system (i.e. DSS-Decision Support System, EIS-Executive Information System and OLAP-Online analytical processing).
- Back-end system: data warehouse, data mart, and data mining [9].

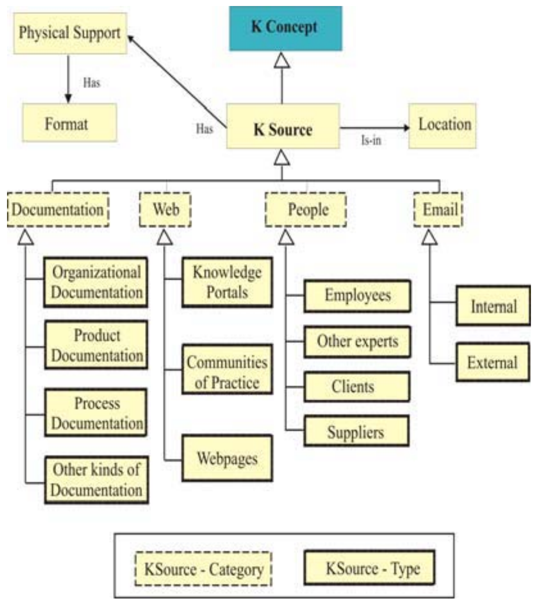


Figure 2. Knowledge Source Ontology

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 [28]. More sophisticated techniques such as webParser [3] to obtain information from the Web, document classification [25], mailing list management [21], 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 representing knowledge topics and knowledge sources [29].

The ontology has four knowledge source categories. These are: 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 finds information is the Web, which can also be divided into other subcategories such as Portals, Communities of Practice, etc. 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, etc. The last knowledge source that the Captor Agent can use is email that can be classified as internal mail (mail sent between employees), and external mail (emails 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.

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 [5]. 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 knowledgebase 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 (proposed in [23]) model is carried out, producing new knowledge resulting in the merging of explicit knowledge plus 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 [10] or REFSENO (Representation Formalism for Software Engineering Ontologies) [33].

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 [20] or on information retrieval techniques ([7], [17]). Other authors such as [32] mix several techniques: data mining and case-based reasoning to develop a recommender system.

Knowledge transfer/application is the most investigated stage in knowledge management [27]. 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, etc. 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 [38]. An appropriate knowledge management linked to communities of practice helps to improve the organization's performance [16]. 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.

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 knowledgebase updated. Therefore, information that is not often used is considered by the Maintenance Agent as information to be possibly eliminated.

3. 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 processes. 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 modelling 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" [4].

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 used for the consistent communication of the agencies.

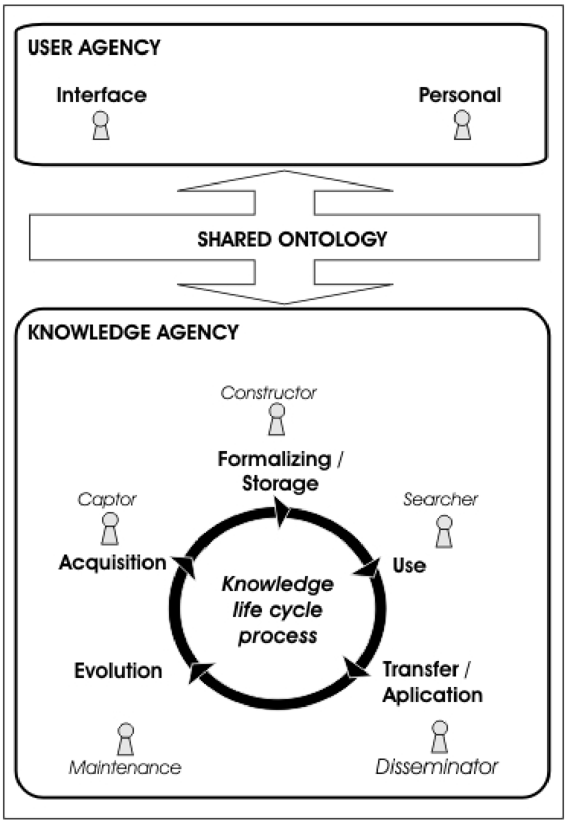


Figure 3. Agents Distribution

In order to carry out the analysis and design of the agents involved we have followed a methodology called INGENIAS [26] 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.

Below, we are going to show two agent meta-model diagrams which describe the roles and tasks of the Captor and Personal agent (one example of each agency). The rest of the agent diagrams are omitted due to space limitations.

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.

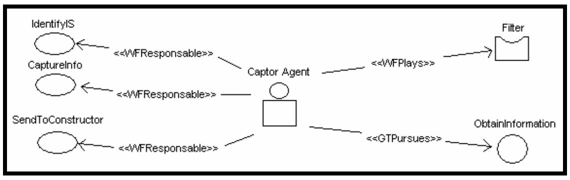


Figure 4. Captor Agent diagram

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.

Figure 5 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
- Recommending information

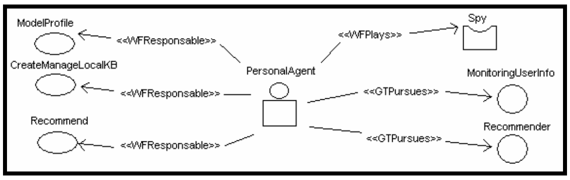


Figure 5. Personal Agent diagram.

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, etc.

CreateManageLocalKnowledgeBase: Creating and managing a local knowledge 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.

4. 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 the maintainers should consult to solve a particular problem. Before constructing the prototype the knowledge flows that take place in software maintenance companies were studied (see [30]). To illustrate how the prototype works let us describe a scenario.

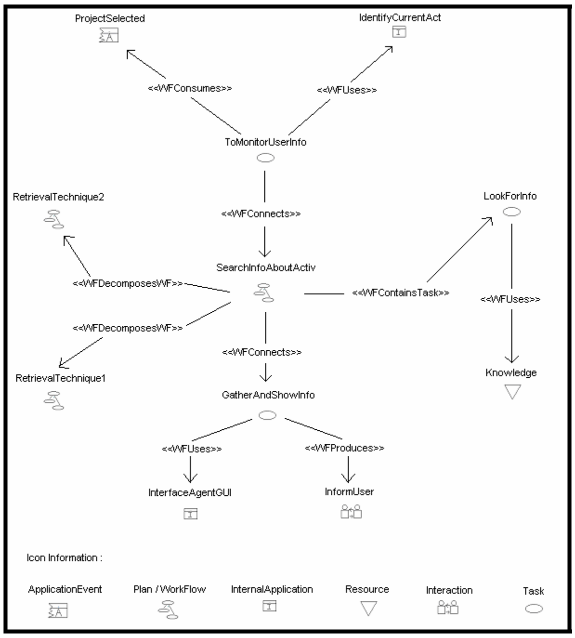


Figure 6. Scenario Diagram

Scenario

A software maintenance engineer selects the project which s/he is going to work on. Then, s/he starts to work on an activity (for instance a maintenance request). At the same time, the Personal Agent is monitoring his/her movements and is logging in what project and activity s/he 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 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 his/her work. The employee will decide if s/he wants to consult this information. Figure 6 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 7 which is an interaction model diagram that the INGENIAS methodology utilizes. These diagrams are very useful to see, at first glance, as agents interact.

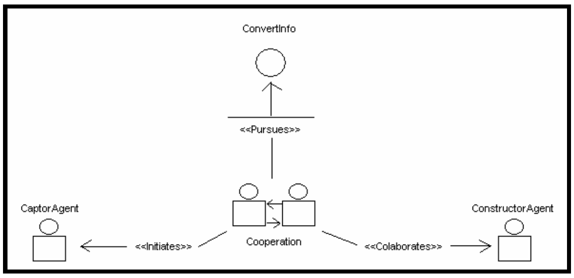


Figure 7. Cooperation between Captor and Constructor Agent

Some Implementation Aspects

The platform that we are using to develop the architecture is JADE (Java Agent Development Framework) 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 [2, 8].

5. 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, [8] (Corporate Memory Management through Agents), which combines emergent technologies, allowing users to exploit an organizational memory.
- To support cooperative activities. For instance in [36] the authors propose a multi-agent architecture to provide support to cooperative activities.
- To recommend. For instance in [32] a system to customize recommendations is described.
- To find experts. Some systems are used to help people find experts which/who can assist them in their daily work .
- To share knowledge. For instance in [19] a multi-agent system is proposed for knowledge sharing in a system designed to advise good programming practice.
- To manage mailing lists , or document classification [21].

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 [11].

6. Conclusions and future work

The main contribution of this paper is the design of a multi-agent model that has been designed to help KMS developers to implement these kinds of systems. The advantages of this model are:

- The model 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 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 model makes use of intelligent agents. These are techniques that have proved to be very convenient in knowledge management activities since they avoid 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 modelling the agents in a systematic way by using INGENIAS methodology whose meta-models 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 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.

Acknowledgement

I wish to thank reviewers for their helpful suggestions to improve this paper.

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

[1] M. Alavi and D. E. Leidner, "Knowledge Management and Knowledge Management Systems: Conceptual Foundations and Research Issues," *MIS Quarterly, 25(1)*, pp. 107-136, 2001.

[2] F. Bergenti, A. Poggi, and G. Rimassa, "Agent Architecture and Interaction Protocols for Corporate Memory Management Systems." In proceedings of the ECAI'2000, Workshop on Knowledge Management and Organizational Memories,2000.

[3] D. Camacho, R. Aler, and J. Cuadrado, "Rule-Based Parsing for Web Data Extraction," in *Intelligent Agents for Data Mining and Information Retrieval*, M. Mohammadian, Ed.: Idea Group, 2004, pp. 65-87.

[4] D. Chin, "User Modelling in UC: The UNIX consultant." In proceedings of the Human Factors in Computing Systems, pp. 24-28,1986.

[5] T. Davenport and L. Prusak, *Working Knowledge: How Organizations Manage What they Know*. Boston, Massachusetts: Harvard Business School Press, pp. 199,1998.

[6] A. Dickinson Enhancing Knowledge Management in Enterprises (ENKE) IST Project, IST-2000-29482, http://www.ist-enke.com, 2000.

[7] W. B. Frakes and R. Baeza-Yates, *Information Retrieval Data Structures and Algorithms*: Englewood Cliffs, NJ: Prentice Hall,1992.

[8] F. Gandon, "A Multi-Agent Architecture for Distributed Corporate Memories." In proceedings of the Third International Symposium From Agent Theory to Agent Implementation in European Meeting on Cybenertics and Systems Research, pp. 623-628,2000.

[9] C. Giannella, R. Bhargava, and K. H., "Multi-Agent Systems and Distributed Data Mining." In proceedings of the Cooperative Information Agents VIII: 8th International Workshop (CIA'04), Erfurt (Germany), pp. 1-15,2004.

[10] T. R. Gruber, "A Translation Approach to Portable Ontology Specification.," *Knowledge Adquisition*, vol. 5(2), pp. 199-220, 1993.

[11] J. Hahn and M. Subramani, "A Framework of Knowledge Management Systems: Issues and Challenges for Theory and Practice." In proceedings of the 21 International Conference in Informatin Systems (ICIS'00), Brisbane,2000.

[12]B. Hansen and K. Kautz, "Knowledge Mapping: A Technique for Identifying Knowledge Flows in Software Organizations." In proceedings of the EuroSPI, pp. 126-137,2004.

[13] N. Houari and B. Homayoun Far, "Application of Intelligent Agent Technology for Knowledge Management Integration." In proceedings of the Third IEEE International Conference on Cognitive Informatics (ICCI'04), pp. 240-249,2004.

[14] M. Jennex, "The Issue of System Use in Knowledge Management Systems." In proceedings of the 38th Hawaii International Conference on System Sciences, Hawaii,2005.

[15] M. Jennex and L. Olfman, "A Model of Knowledge Management Success." International Journal of KNowledge Management, 2 (3), pp. 51-68, 2006.

[16] E. Lesser and J. Storck, "Communities of practice and organizational performance," *IBM Systems Journal*, vol. 40, pp. 831-841, 2001.

[17] T. Liang and J. S. Huang, "A framework for Applying Intelligent Agents to Support Electronic Trading", *Decision Support Systems*, vol. 28(4), pp. 305-317, 2000.

[18] P. Maes, "Agents that reduce work and information overload," *Communications of the ACM*, vol. 37, pp. 31-40, 1994.
[19] S. Mercer and S. Greenberg, "A Multi-Agent Architecture for Knowledge Sharing". In proceedings of the Sixteenth European Meeting on Cybernetic and Systems Research, Viena,2001.
[20]M. Mohammadian and R. Jentzsch, "Computational Intelligence Techniques Driven Intelligent Agents for Web Data Mining and Information Retrieval", in *Intelligent Agents for Data Mining and Information Retrieval*, M. Mohammadian, Idea Group, 2004.
[21] E. Moreale and S. Watt, "An Agent-Based Approach to Mailing List Knowledge Management." In proceedings of the Agent-Mediated Knowledge Management, pp. 118-129,2003.
[22] M. Nissen, "An Extended Model of Knowledge-Flow Dynamics," *Communica-tions of the Association for Information Systems*, pp. 251-266, 2002.
[23] I. Nonaka, "A Dynamic Theory of Organizatinal Knowledge Creation," *Organization Science*, vol. 5:1, pp. 14-37, 1994.
[24] I. Nonaka and H. Takeuchi, *The Knowledge-Creating Company*: Oxford University Press, pp. 304,1995.
[25] J. Novak, M. Wurst, M. Fleschmann, and W. Strauss, "Discovering, Visualizing and Sharing Knowledge through Personalized Learning Knowledge Maps". In proceedings of the Agent-Mediated Knowledge Management, Standford (USA), pp. 213-228,2003.
[26] J. Pavón and J. J. Gómez-Sanz, "Agent Oriented Software Engineering with INGENIAS". In proceedings of the Multi-Agent Systems and Applications III: CEEMAS 2003 (LNAI), Prage, Czech Republic, pp. 394-403,2003.
[27] T. Peachey and D. Hall, C. Cegielski "Knowledge Management and the Leading Information Systems Journals: An Analysis of Trends and Gaps in Published Research." International Journal of Knowledge Management, 1 (3), pp. 55-69, 2005.
[28] A. J. Rhem, *UML for Developing Knowledge Management Systems*. New York: Auerbach Publications, 2006.
[29] O. Rodríguez, A. I. Martínez, J. Favela, A. Vizcaíno, and M. Piattini, "Understanding and Supporting Knowledge Flows in a Community of Software Developers." In proceedings of the X International Workshop on Groupware (CRIWG 2004) LNCS 3198, Springe, San Carlos, Costa Rica, pp. 52-66, 2004.
[30] O. Rodríguez, A. I. Martínez, A. Vizcaíno, J. Favela, and M. Piattini, "Identifying Knowledge Flows in Communities of Practice," in *Encyclopedia of Communities of Practice in Information and Knowledge Management*, E. Coakes and S. A. Clarke, Eds.: Idea Group, 2005.
[31] I. Rus and M. Lindvall, "Knowledge Management in Software Engineering," *IEEE Software*, vol. May/June, pp. 26-38, 2002.
[32] J. Sung Kim, "Customized Recommendation Mechanism Based on Web Data Mining and Case-Based Reasoning," in *Intelligent Agents for Data Mining and Information Retrieval*, M. Mohammadian, Ed., 2004.
[33] C. Tautz and C. Von Wangenheim, "REFSENO: A Representation Formalism for Software Engineering Ontologies. Fraunhofer IESE-Report Nº 015.98/E, version 1.1 October 20," 1998.
[34] A. Tiwana, *The Knowledge Management Toolkit: Practical Techniques for Building a Knowledge Management System*. USA: Prentice Hall, pp. 608,2000.
[35] L. van Elst, V. Dignum, and A. Abecker, "Agent-Mediated Knowledge Management." In proceedings of the Agent-Mediated Knowledge Management, Standford (USA), pp. 1-30,2003.
[36] A. Wang, C. Reidar, and L. Chunnian, "A Multi-Agent Architecture for Cooperative Software Engineering". In proceedings of the Eleventh International Conference on Software Engineering and Knowledge Engineering (SEKE'99), Germany, pp. 162-169,1999.
[37] J. Ward, and Aurum, A., "Knowledge Management in Software Engineering - Describing the Process". In proceedings of the Proceedings of Australian Software Engineering Conference, ASWEC'04, Melbourne, Australia, 2004.
[38] E. Wenger, *Communities of Practice: Learning, Meaning, and Identity*: Cambridge, U.K.: Cambridge University Press, 1998.
[39] K. Wiig, "Knowledge Management: Where did it come from and wher will it go?," *Expert Systems with Applications*, vol. 13, pp. 1-14, 1997.
[40] M. Wooldridge and N. R. Jennings, "Intelligent Agents: Theory and Practice," *The Knowledge Engineering Review*, vol. 10, (2), pp. 115-152, 1995.

Appendix One

Model	Stage1	Stage2	Stage3	Stage4	Stage5	Stage6	Stage7
Nonaka y [24] Takeuchi	Socialization	Externalization	Combination	Internalization			
Wiig [39]	Creation	Storing/ gathering	Use	Leverage	Sharing		
Davenport y Prusak [5]	Generation	Codify/ Coordinate	Transfer	Roles and Skills			
Tiwana [34]	Acquire	Sharing	Use				
Alavi y Leidner [1]	Creation	Storage/ Retrieval	Transfer	Application			
Rus y Lindvall [31]	Creation/ Acquisition	Organization/ Storage	Distribution	Application			
Nissen [22]	Creation	Organization	Formalize	Distribute	Application	Evolve	
Ward y Aurum [37]	Creation	Distribution	Organization	Adaptation	Identification	distribution	Application
Dickinson [6]	Identification	Acquisition	Development	Distribution	Use	Preservation	