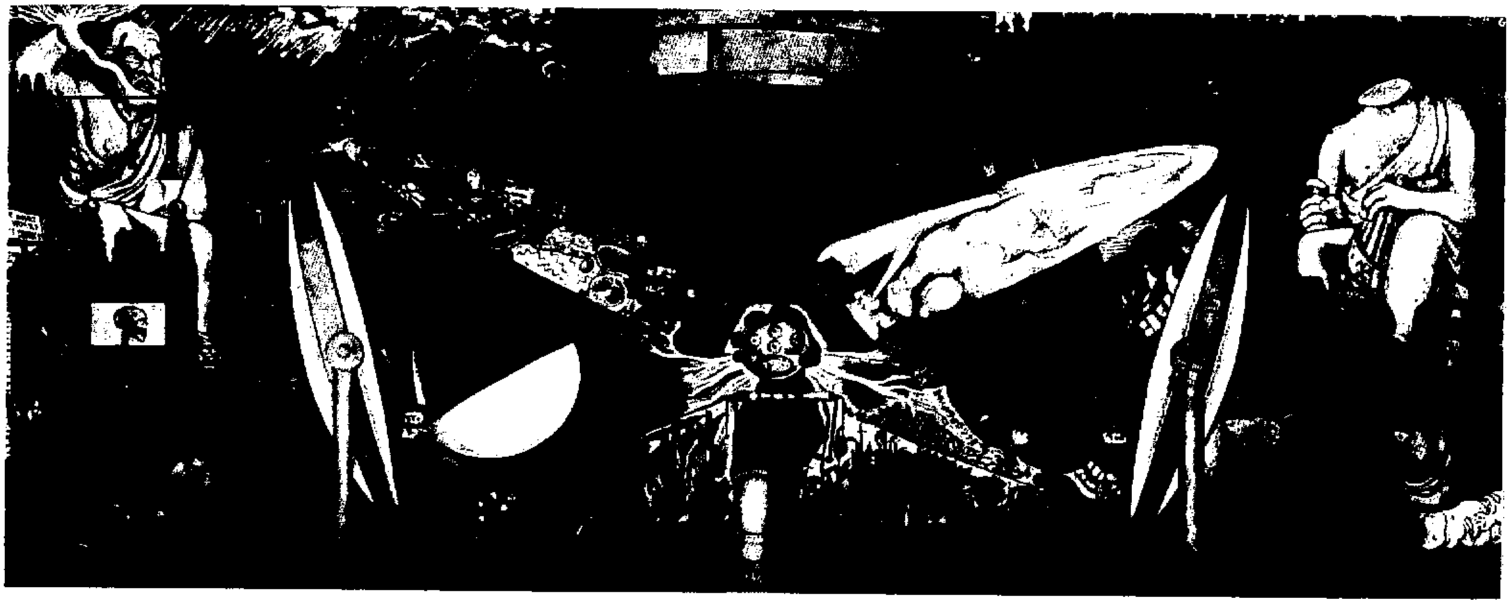
15th International Conference on Computing
CIC-2006

November 21-24, 2006 Mexico City, Mexico



MAGNO 2006

Edited by
Alexander Gelbukh
Sergio Suárez Guerra

70
Aniversario
Instituto Politécnico Nacional
1936-2006

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

IEEE Computer Society Order Number P2708
Library of Congress Number 2006931165
ISBN 0-7695-2708-6

ISBN 0-7695-2708-6
90000
9 780769 527086

Proceedings

15th International Conference on Computing

CIC 2006

21-24 November 2006 • Mexico City, Mexico

Edited by
Alexander Gelbukh and Sergio Suárez Guerra

IEEE COMPUTER SOCIETY
IEEE
Los Alamitos, California
Washington • Tokyo

IEEE Computer Society Order Number P2708
ISBN-13: 978-0-7695-2708-6
ISBN-10: 0-7695-2708-6
Library of Congress Number 2006931165

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

Editorial production by Lisa O'Conner

Cover art production by Joe Daigle/Studio Productions

Printed in the United States of America by Applied Digital Imaging

IEEE COMPUTER SOCIETY

IEEE

IEEE Computer Society
Conference Publishing Services
http://www.computer.org/proceedings/

Table of Contents

15th International Conference on Computing

CIC 2006

ARTIFICIAL INTELLIGENCE

Knowledge Management, Reasoning, Neural Networks, and Evolutionary Programming

Image Processing and Pattern Recognition

---

* **Best Paper Award, 3rd Place**

Machine Learning and Data Mining

Natural Language Processing and Speech Recognition

Information Retrieval

---

† **Best Paper Award, 2nd Place**

Multi-Agent Systems and Ontologies

Bioinformatics and Medical Applications

Intelligent Tutoring Systems

COMPUTER SCIENCE

Formal Languages and Automations

---

‡ **Best Student Paper Award**

Software Engineering and Data Warehousing

Cryptography and Security

Computer Networks and Distributed Systems

§ **Best Paper Award, 1st Place**

Mobile Computing

Control

Preface

This volume contains the papers presented at the oral session of the 15th International Conference on Computing held on November 21–24, 2006, in Mexico City, Mexico. The conference was part of the Magnum Conference on Computing organized by the Center for Computing Research of the National Polytechnic Institute of Mexico.

During the years passed from its first edition, the International Conference on Computing has gradually become the largest national and a major international conference on computer science. Starting from this year's event, the proceedings of the conference will be published by the IEEE Computer society. With this conference we celebrated several important events: the 70th anniversary of the National Polytechnic Institute of Mexico, the 60th anniversary of the IEEE Computer society, the 15th anniversary of the International Conference on Computing series, and the 10th anniversary of the Center for Computing Research.

The number of submissions to CIC-2006 session was higher than in the previous years: 198 papers by 510 authors from 30 different countries where submitted for evaluation, see Tables 1 and 2. The submissions were reviewed by Program Committee members and additional reviewers. This volume contains revised versions of 63 papers by 168 authors selected after a thorough evaluation for presentation at the oral session. Thus the acceptance rate was 31.8%.

The volume is structured into 14 thematic fields representative of the main current areas of interest within the Artificial Intelligence and Computer Science community:

- Knowledge Management, Reasoning, Neural Networks, and Evolutionary Programming,
- Image Processing and Pattern Recognition,
- Machine Learning and Data Mining,
- Natural Language Processing and Speech Recognition,
- Information Retrieval,
- Multi-Agent Systems and Ontologies,
- Bioinformatics and Medical Applications,
- Intelligent Tutoring Systems,
- Formal Languages and Automations,
- Software Engineering and Data Warehousing,
- Cryptography and Security,
- Computer Networks and Distributed Systems,
- Mobile Computing,
- Control

In addition to the oral session, the conference featured a poster session, short tutorials, exhibitions, and keynote presentations. The participants attended excellent keynote talks presented by the invited speakers of the Magnum Conference on Computing:

- Jesús Cardeñosa Lera, Polytechnic University of Madrid: *Universal Networking Language*;
- César Galindo Legaria, Microsoft Corporation: *Evolution of Database Technologies*;
- Armando Guevara, Gtt NetCorp, Inc.: *Is-Net.Net: Spacially-enabled Information Services*;
- Manuel Reyes Gómez, Microsoft Research: *Probabilistic Graphical Models for Audio and Biology*;
- Juan Eduardo Vargas, Univ. of South California: *Data Mining*;
- Alexander Veidenbaum, Univ. of California at Irvine, *Architectural Technologies for Reducing Energy Consumption in Associative Data Caches*;
- Bernard Widrow, Stanford University. *Cognitive Memory and its Applications.*

The following papers were judged by the Program Chairs as the best papers of the conference:

Best Paper Award, 1st place: *A Model and Language for Bitemporal Schema Versioning in Data Warehouses,* by Ericka-Janet Rechy-Ramírez and Edgard Benítez-Guerrero from Mexico;

Best Paper Award, 2nd place: *A Maximum Entropy Model Application on Recognition of Metaphor Phenomena,* by Yang Xu from China;

Best Paper Award, 3rd place: *Robust Estimation of Background for Fixed Cameras,* by Aristodemos Pnevmatikakis and Lazaros Polymenakos from Greece;

Best Student Paper Award: *Modelling a Knowledge Management System Architecture with INGENIAS Methodology,* by Juan P. Soto, Aurora Vizcaíno, Javier Portillo, and Mario Piattini from Spain.

Table 1. Statistics of submissions and accepted papers by country / region

Country	Authors		Papers[1]		Country	Authors		Papers[1]	
	Subm	Accp	Subm	Accp		Subm	Accp	Subm	Accp
Algeria	1	–	1	–	Ireland	3	2	2	1
Austria	4	4	1	1	Israel	2	–	1	–
Bangladesh	1	–	0.33	–	Japan	5	1	2	1
Brazil	12	5	5	2	Korea, South	14	8	5	2
Cameroon	11	–	2.35	–	Latvia	2	–	1	–
Canada	13	2	5	1	Mexico	237	84	88.5	30
Chile	1	–	0.25	–	Norway	1	–	0.5	–
China	67	15	27.5	6.5	Pakistan	3	–	2	–
Cuba	2	–	1	–	Portugal	20	9	6.75	3
France	25	4	12.25	2.5	Romania	1	1	0.33	0.33
Germany	3	–	1.15	–	Spain	26	16	8	5.13
Greece	6	6	2	2	Taiwan	3	–	1	–
Iceland	1	–	0.2	–	UK	7	2	3.25	1
India	20	2	7.8	1	USA	14	7	5.75	3.5
Iran	4	–	2	–	Venezuela	1	–	1	–
					Total:	*510*	*168*	*198*	*63*

[1] By authors: e.g., for a paper by 2 authors from UK and 1 from USA, we added ⅔ to UK and ⅓ to USA.

The best student paper was selected out of the papers of which the first author was a full-time student.

We want to cordially thank all people involved in the organization of this conference. In the first place, these are the authors of the papers constituting this book: it is the excellence of their research work that gives value to the book and sense to the work of all other people involved. We thank the members of the Program Committee and additional reviewers for their hard work on selecting the best papers to be included in this volume. We also thank the conference staff and the members of the Organizing Committee, who carried out the great effort of preparation of the conference with professionalism and enthusiasm without which this conference would not have been possible and this book would not have appeared. We thank Sulema Torres and Hiram Calvo for their help in the preparation of this volume and in reviewing process. The entire submission, reviewing, and selection process, as well as putting together the proceedings, was supported for free by the EasyChair system; we express our gratitude to its author Andrei Voronkov for his constant support and help. Last but not least, we deeply appreciate the patience and help of the IEEE staff, in particular Lisa O'Conner, in editing this volume. The conference was funded by Mexican Government through National Polytechnic Institute.

Alexander Gelbukh
Sergio Suárez Guerra
Program Chairs

October 2006

Table 2 Statistics of submissions and accepted papers by topic[2]

Accepted	Submitted	% accepted	Topic
10	15	67	Information Retrieval
8	14	57	Machine Learning and Data mining
7	18	39	Modeling and Simulation
6	9	67	Multi-Agent Systems and Distributed AI
6	18	33	Pattern Recognition
5	11	45	Knowledge Representation and Knowledge-Based Systems
5	7	71	Natural Language Processing, Ontologies, and Semantic Web
4	5	80	Automations, Formal Languages and Grammars
4	14	29	Bioinformatics and Medical Applications
4	11	36	Educational Software
4	17	24	Image Processing
4	8	50	Network Analysis and Management
4	11	36	Network Architectures and Protocols
4	13	31	Ubiquitous and Mobile Computing
4	18	22	Web Services
3	8	38	Fuzzy Logic, Probabilistic and Uncertainty Reasoning
3	6	50	Graph and Discrete Mathematics algorithms
3	8	38	Human-computer interfaces
3	6	50	Intelligent Tutoring Systems
3	6	50	Interconnectivity
3	15	20	Programming Languages and Techniques
3	6	50	Software System Modeling and UML
2	6	33	Database Technology and Data Warehousing
2	4	50	Decision Support Systems
2	5	40	Intelligent e-Commerce, e-Government, and e-Anything
2	3	67	Interoperability
2	3	67	Intrusion Detection and Deception Systems
2	5	40	Logic, Reasoning, Planning, and Constraint Programming
2	11	18	Mathematical methods in control
2	8	25	Naturally-Inspired Computing and Evolutionary Algorithms
2	10	20	Optimization algorithms
2	4	50	Software Reuse and Lifetime Cycle
2	3	67	Speech and voice processing
2	7	29	Wavelets Analysis
2	11	18	Wireless networking technology
1	2	50	Biometrics and Physical Security
1	8	13	Compilers
1	8	13	Computer Vision
1	1	100	Digital Signal Processing Algorithms
1	6	17	Embedded Systems
1	4	25	GRID and cluster computing
1	4	25	Hybrid Intelligent Systems
1	4	25	Intelligent Control Systems
1	7	14	Intelligent Decision Making
1	2	50	Reverse Engineering
1	1	100	Trajectory Planning
–	1	–	Antispam and Antivirus Technology
–	2	–	Computer Architecture
–	2	–	Cryptography and Cryptanalysis
–	1	–	Data compression
–	2	–	Flexible Manufacturing
–	1	–	Low Power Consumption Devices
–	3	–	Mobile Robotics
–	4	–	Operating Systems
–	1	–	Processor Architecture
–	3	–	Secure Network Architectures and Protocols
–	3	–	Security in Electronic Commerce

[2] Topics indicated by the authors. A paper may be assigned to more than one topic.

Program Committee

Ajith Abraham	Chung-Ang University, Korea, South
Khaldoun Al Agha	Laboratoire de Recherche en Informatique, France
Scott Ambler	IBM, Canada
John Atkinson	Universidad de Concepcion, Chile
Iain Bate	University of York, United Kingdom
Nieves Brisaboa	Universidade da Coruña, Spain
Klaus Brunnstein	University of Hamburg, Germany
Andre C. P. L. F. de Carvalho	Universidade de São Paulo, Brazil
John Debenham	University of Technology, Sydney, Australia
Jürgen Dix	Clausthal University of Technology, Germany
Abdennour El Rhalibi	Liverpool John Moores University, United Kingdom
Juan Flores	Michoacan University, Mexico
Matjaž Gams	Jozef Stefan Institute, Slovenia
Alexander Gelbukh	National Polytechnic Institute, Mexico, co-chair
Adolfo Guzmán Arenas	National Polytechnic Institute, Mexico
Sang Yong Han	Chung-Ang University, Korea, South
Hans-Ludwig Hausen	Fraunhofer, Germany
Heinz-Gerd Hegering	Leibniz-Rechenzentrum, Germany
Dieter Hutter	DFKI, Germany
Mario Köppen	Kyushu Institute of Technology, Japan
Eugene Levner	HIT, Israel
Antonio Liotta	University of Essex, United Kingdom
Peter Müller	ETH Zurich, Switzerland
Raul Monroy	ITESM-CEM, Mexico
Alberto Ochoa Rodríguez	ICIMAF, Cuba
Alex Ramírez	Universitat Politècnica de Catalunya, Spain
Isidro Ramos	Universitat Politècnica de Valencia, Spain
Fuji Ren	University of Tokushima, Japan
Carlos Alberto Reyes	INAOE, Mexico
Roberto Rodríguez Morales	Institute of Cybernetics, Mathematics & Physics, Cuba
José Ruiz Shulcloper	Advanced Technology Applications Center, Cuba
Michelle Sibilla	Paul Sabatier University, IRIT, France
Grigori Sidorov	National Polytechnic Institute, Mexico
Sergio Suárez Guerra	National Polytechnic Institute, Mexico, co-chair
Andrei Tchernykh	CICESE, Mexico
Manuel Vilares	University of Vigo, Spain
Andrei Voronkov	University of Manchester, United Kingdom
Franz Wotawa	Technische Universität Graz, Austria
Berend Jan van der Zwaag	University of Twente, Netherlands

Additional Reviewers

J.C. Acosta Guadarrama
Carlos Aguilar Ibañez
Bibiane Angerer
Nils Bulling
Hiram Calvo Castro
José Ramón Calvo de Lara
Victor Manuel Darriba Bilbao
Osvaldo Espinosa Sosa
Mario Estrada
Mohamed Abdel Fattah
Edgardo Manuel Felipe Riverón
Karen A. García Gamboa
Lucina Garcia
Edel Garcia Reyes
Fernando Godinez
Abel Gómez
Miguel Gonzalez
Agustín F. Gutiérrez Tornés
Jorge Herrerías
Xiao Hui Jin
Emilio Insfran
Hermann Lehner
Patricio Letelier
Juan Carlos López Pimentel
Fernando Magán Muñoz
Tristan Marc Behrens
Jesús Alberto Martínez Castro
José Francisco Mata Silva
Jaime Mora
Elena Navarro
Martin Nordio
Peter Novak
José Ramón Paramá
Bernhard Peischl
Miguel R. Penabad
Aurora Pérez
Jennifer Pérez
Angeles S. Places
Leticia Flores Pulido
Oleksiy Pogrebnyak
Jose Federico Ramirez
Orion F. Reyes Galaviz
Francisco José Ribadas Pena
Elsa Rubio Espino
Arsenii Rudich
Joseph N. Ruskiewicz
Pedro Sánchez Palma
Francisco José Silva Mata
Marta Soto
Hind Taud
Carlos Vega
Jesus Vilares

Organizing Committee

Program Chairs	Alexander Gelbukh Sergio Suárez Guerra
Finance Chair	Raquel López Alamilla
Sponsorship Chair	Alfredo Viniegra Islas
Logistics and Publicity Chair	Oralia del Carmen Pérez Orozco
Technical Support Chair	Martín Haro Martínez
Editorial Support	Sulema Torres Ramos
Members of Organizing Committee	César Saul Guzmán Rentería Joaquín Medina Sergio Sandoval Reyes

Modelling a Knowledge Management System Architecture with INGENIAS Methodology

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

Abstract

Nowadays knowledge management is a topic of vital importance as companies have realized that it offers a competitive advantage. In order to develop a knowledge management system that helps companies to take advantage of their knowledge, different technologies have been used. One of those which has proved to be rather efficient is that of intelligent agents, since these agents can develop tasks on behalf of users. Moreover, agents may use different intelligent techniques and communicate with each other.

Taking this into account we are developing a multi-agent architecture to help developers to implement knowledge management systems. Before designing the architecture we studied what agent methodology was the most suitable to carry out this task. After a preliminary study we consider that MASCommonKADS and INGENIAS could be the most convenient. In this paper we describe why we finally chose INGENIAS to design our architecture. Moreover, the design of the architecture using the INGENIAS model is also described.

1. Introduction

In recent years, knowledge has become a very important factor; in fact, intellectual capital is one of the most important assets for many organizations [12]. Because of this, topics such as Knowledge Management (KM) are currently of special interest to organizations who are worried about their employees' learning and competitiveness since a suitable management of knowledge can help organizations to increment the collaboration of their members and encourage the sharing of knowledge between them. There are many definitions of KM. For example, KM can be defined as a discipline that enables an organization to take advantage of its intellectual capital in order to reuse it and learn from previous experience [20]. In [21] Skyrme suggests that KM is the explicit and systematic management of vital knowledge and its associated processes of creating, gathering, organizing, diffusing, using, and exploiting that knowledge. Due to the importance of KM, tools which support some of the tasks related to KM have been developed. Different techniques are used to implement KM systems. One of them, which is providing to be quite useful, is that of intelligent agents [23, 24]. Software agent technology can monitor and coordinate events, meetings and disseminate information [1]. 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 the work that employees have to perform when using a KM system. Another important issue is that agents can learn from their own experience. The advantages that agent technology has shown in the area of information management encouraged us to consider agents as a suitable technique to develop an architecture with the goal of helping to develop KM systems.

Once the architecture had been chosen, the next step was to select a methodology to develop it. Nowadays, there are many methodologies to develop multi-agent systems, in this paper we describe why INGENIAS was chosen and how we have followed this methodology to develop our architecture.

The remainder of the paper is organized as follows. In section two we briefly describe two methodologies for the construction of multi-agent systems: MAS-CommonKADS and INGENIAS. Moreover the methodologies are compared and the advantages and disadvantages of each one are explained. Later, in section three it is describe how we are using the INGENIAS methodology to develop our architecture.

0-7695-2708-6/06 $20.00 © 2006 IEEE

Finally, conclusions and future work are outlined in section four.

2. Methodologies for developing multi-agents systems

More and more effort is made to provide methodologies for the development of agent based systems. In literature we found several agent-oriented methodologies, such as: KAOS [2], DESIRE [3], AUML [16], MESSAGE [5], MaSE [10], TROPOS [4], GAIA [26], MAS-CommonKADS [14] and INGENIAS [17] among others. These methodologies are focused on providing support to the each one of the phases of the development process of an agent oriented system. Other authors have focused their work on making comparative of the diverse methodologies proposed. For instance, in [8] Dam and Winikoff propose a framework to compare Agent Oriented Software Engineering (AOSE) methodologies. In [6] Cernuzzi and Rossi propose a framework to carry out an evaluation of agent-oriented analysis and design modelling methods.

After the study of literature about different agent oriented methodologies we considered that the two most suitable methodologies for our work were two. INGENIAS and MAS-CommonKADS for the following reasons: MAS-CommondKADS is related to knowledge management and INGENIAS is considered by many authors as one of the most up to date and complete methodology.

Next we shall describe the main characteristics of each one of them.

2.1. MAS-CommonKADS

This methodology is called MAS-CommonKADS because extends CommonKADS (Knowledge Acquisition and Design Structuring) methodology, adding the aspects which are relevant to MAS. CommonKADS [9] is a methodology designed for the development of knowledge base system (KBS) analogous to methods of software engineering. The development of this methodology has been funded by European Community's ESPIRIT program between 1983 and 1994.

MAS-CommonKADS defines the following models [14]: 1) Agent Model, that describes the characteristics of each agent; 2) Task Model, that describes the tasks that the agents carry out; 3) Expertise Model, that describes the knowledge needed by the agents to achieve their goals; 4) Organization Model, that describes the structural relationships between agents (software agents and/or human agents); 5) Coordination Model, that describes the dynamic relationships between software agents; 6) Communication Model, that describes the dynamic relationships between human agents and their respective personal assistant software agents; and 7)Design Model, that refines the previous models and determines the most suitable agent architecture for each agent, and the requirements of the agent network.

The development process followed by MAS-CommonKADS follows these phases: 1) Conceptualization, 2) Analysis, 3) Design, 4) Coding and testing, Integration, and 6) Operation and maintenance described in [13]. MAS-CommonKADS is focused on the knowledge, tasks and reasoning techniques used by agents. Therefore it allows to model to the knowledge and reasoning of the agents.

2.2. INGENIAS

INGENIAS [17] is an agent oriented software engineering methodology for multi-agent systems development; it reuses ideas from MESSAGE/UML [5], a methodology which has based its notion on UML.

In INGENIAS, the general approach to specify MAS is to divide the problem into more concrete aspects that form different view of the system. The difference of this proposal from existing ones is how these views are defined and built, and how they are integrated in the MAS development.

Each type of view is described using a meta-modeling language which specifies a kind of grammars with which we build models, known as views. INGENIAS recognizes five meta-models that describe systems views. Each view, being the instance of a meta-model, is named model. So to describe a MAS, the developer must specify following models [11]:

- Agent model. It describes single agents, their tasks, goals, initial mental state, and played roles.
- Interaction model. Describes how interaction among agents takes place.
- Tasks and goals model which describes relationships among goals and tasks, goal structures, and task structures.
- Organization model to define how system components (agent, roles, resources and applications) are grouped together, which tasks are executed in common, which goals they share, and what constraints exists in the interaction among agents.
- Environment model which defines agent's perception in terms of existing elements of the system. It also identifies system resources and who is responsible of their management.

2.3. Comparing methodologies

Diverse perspectives have been given to each one of the described methodologies. For instance, MAS-CommonKADS was developed focused on the perspective from a knowledge-based systems approach, and others as an evolution of object-oriented systems.

Based on the above methodologies, we have developed an analysis to identify the advantages of each methodology. First, we concluded that INGENIAS provides the best support in all the phases of the lifecycle of an agent-based application, including its management. Second, INGENIAS provides tools support that MAS-CommonKADS has not yet. For instance, INGENIAS provides a visual language for multi-agent systems definition, a process to guide the lifecycle of the software development based on the Rational Unified Software Development Process (RUP). In addition, the code generator itself is another tool developed for INGENIAS. It produces code according to the instructions given by the specification of the system. With this tool is possible to generate code no matter what agent platform is being used [18].

For these reasons we decided to use INGENIAS methodology which is one of the most complete methodologies for the development of multi-agent systems, since it proposes a very descriptive language, an easy to follow process, models that can be constructed with different levels of abstraction and from different views or perspectives, and a tool based on meta-models. Using meta-models facilitates the development of the system enormously, as it will be illustrated in next section, since they are oriented towards visual representations of concrete aspects of the system and the tool can automatically check the meta-models. Finally, INGENIAS define the models used to describe a MAS in more detail. Nevertheless, MAS-CommonKADS includes an expertise model related to the knowledge management that INGENIAS doesn't have. The expertise model is used for modelling the reasoning capabilities of the agents to carry out their tasks and achieve their goals.

3. Multi-agent architecture

In the next paragraphs we illustrate how we have used INGENIAS to develop our architecture.

The architecture has two main types of agencies: knowledge agency and user agency (Figure 1). Next, we are going to explain how the agents are structured into the two agencies with the goal of joining all the agents closely in charge of managing knowledge and supporting the different stages of the knowledge model proposed in [22].

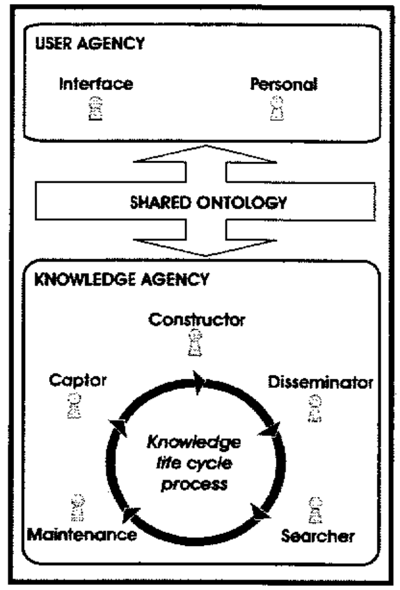


Figure 1. Multi-agent architecture

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.

- *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 a knowledge ontology which defines the knowledge to be taken into account in a domain.
- *Constructor 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 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.

- *Searcher Agent* is in charge of gives recommendations or suggestions with the goal of helping users to perform their tasks by reusing lessons already learnt as well previous experiences. This agent could be implemented with different retrieval techniques. Since this architecture is proposed at a high level these aspects will not be dealt with in this paper.
- *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 [25]. An appropriate knowledge management linked to communities of practice helps to improve the organization's performance [15]. 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 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.
- *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.

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" [7].

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.

3.1 Modelling the system architecture

In this section we are going to show two agent meta-model diagrams which describe the roles and tasks of the Captor, Personal and Constructor agent. The rest of the agent diagrams are omitted due to space limitations.

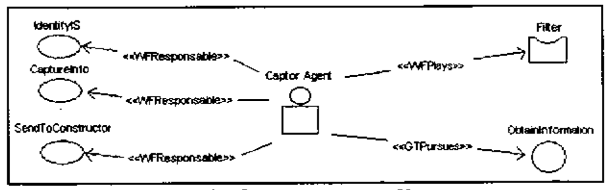


Figure 2. Captor agent diagram

Figure 2 shows that the goal of this 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 information 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.

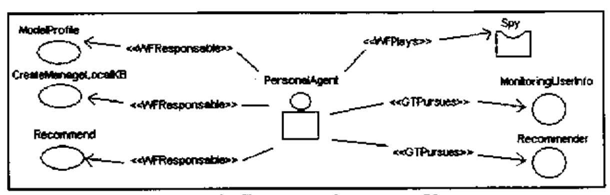


Figure 3. Personal agent diagram

Figure 3 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:

Modelling 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 information would be relevant for the user. To accomplish this, this agent communicates with the Searcher agent and with the Interface agent.

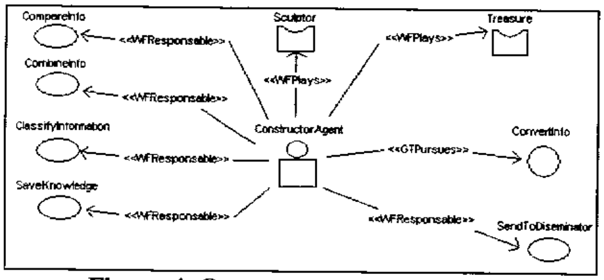


Figure 4. Constructor agent diagram

Figure 4 shows the role and tasks performed by the Constructor Agent 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 knowledgebase (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. In this way, the combination process of the SECI model is carried out, producing new knowledge resulting in the merging explicit knowledge plus explicit knowledge.

ClassifyInformation: Another task is to classify the information received by the Captor Agent (for instance: models, structures, files, diagrams, etc.).

SaveKnowledge: One of the most important tasks is to store the new knowledge into the knowledgebase.

SendToDisseminator: This is a critical task which consists of sending knowledge to the Disseminator Agent.

3.2 Modelling agents interaction

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 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 [19]). To illustrate how the prototype works let us describe a scenario.

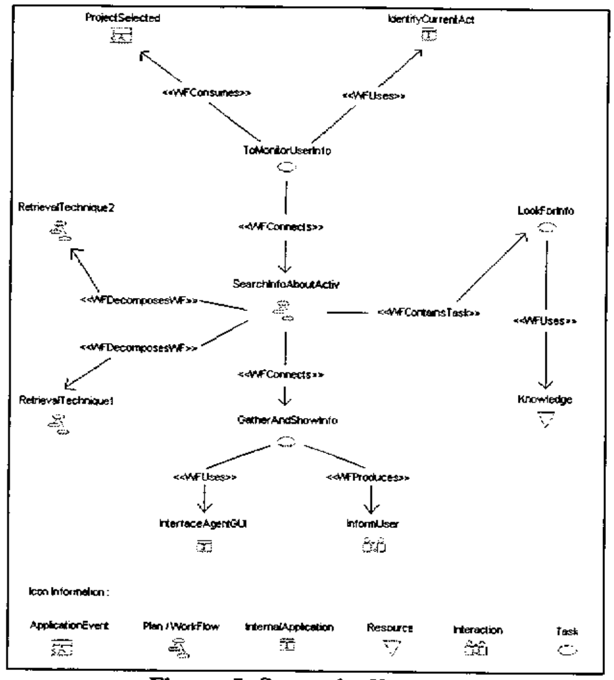


Figure 5. Scenario diagram

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 5 depicts the diagram of this process.

Once, the employee has finished the work the Captor Agent checks whether a new case can be constructed (in case the employee has found a solution to a problem) or whether a new knowledge source has been used. In both cases the Captor sends the new information 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 circumstances that have taken place.

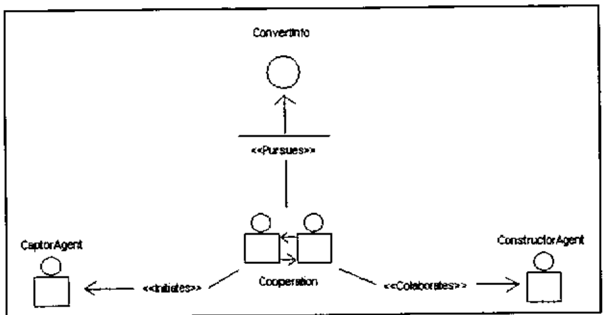


Figure 6. Cooperation between captor and constructor agent

The collaboration between the Captor and the Constructor Agent is depicted in Figure 6 which is an interaction model diagram that the INGENIAS methodology utilises. These diagrams are very useful to see, at first glance, how agents interact.

Some implementation aspects

The platform that we are using to develop the architecture is 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. In association with JADE we are using Jess, a rule engine and scripting environment. Jess, can be used to build software that has the capacity to "reason". It works by using knowledge supplied in the form of declarative rules.

4. Conclusion and future work

This paper compares briefly two methodologies that a according to our study are the most mutable to develop a KM system. We concluded that INGENIAS is one of the most complete methodologies for the development of MAS due to provide the best support in all the phases of the lifecycle of an agent-based application, including its management.

The main contribution of this work is to describe how the methodology finally chosen (INGENIAS) was used to develop our architecture.

As future work we aim to compare the implementation of a KMS based on our proposal with developments using other architectures.

5. Acknowledgement

This work is partially supported by the ENIGMAS (PIB-05-058), MECENAS (PBI06-0024) and ESFINGE (TIN2006-15175-C05-05) project, Junta de Comunidades de Castilla-La Mancha, Consejería de Educación y Ciencia, both in Spain.

6. References

[1] Balasubramanian, S., Brennan, R., Norrie, D, "An Architecture for Metamorphic Control of Holonic Manufacturing Systems", *Computers in Industry*, Vol. 46, no. 1, pp. 13-31, (2001).

[2] Bradshaw, J., "KAoS: An Open Agent Architecture Supporting Reuse, Interoperability, and Extensibility", *Knowledge Acquisition Workshop* (KAW), (1996).

[3] Brazier, F., Dunin-Keplicz, B., Jennings, N., and Treur, J., "DESIRE: Modelling Multi-Agent Systems in a Compositional Formal Framework", *International Journal of Cooperative Information Systems,* special issue on Formal Methods in Cooperative Information Systems: Multi-Agent Systems, (1997).

[4] Bresciani, P., Perini, A., Giorgini, P., Giunchiglia, F., and Mylopoulus, J., "Knowledge Level Software Engineering Methodology for Agent Oriented Programming", *Proceedings of the Fifth International Conference on Autonomous Agents, ACM,* Montreal Canada, (2001).

[5] Caire, G., Leal, F., Chainho, P., Evans, R., Gómez-Sanz, J., Pavón, J., Kearney, P., Stark, J., and Massonet, P., "MESSAGE: Methodology for Engineering Systems of Software Agents", *Technical Information - European Institute for Research and Strategic Studies in Telecommunications* (EURESCOM), (2001).

[6] Cernuzzi, L., and Rossi, G., "On the Evaluation of Agent Oriented Modelling Methods", *Proceedings of Agent Oriented Methodology Workshop.* Seattle USA, (2002).

[7] Chin, D., "User Modelling in UC: The UNIX consultant". *Human Factors in Computing Systems.* (1986).

[8] Dam, K.H., and Winikoff, M., "Comparing Agent-Oriented Methodologies", *Proceedings of the 5th International Bi-Conference Workshop on Agent Oriented Information Systems (AOIS)*, Melbourne, Australia (2003).

[9] de Hoog, R., Martil, R., Wielinga, B., Taylor, R., Bright, C., and van de Velde, W., "The CommonKADS model set", *ESPIRIT Project P5248 KADS-II/M1/DM. 1b/UvA/018/5.0.*: University of Amsterdam, Lloyd's Register, Touche Ross Management Consultants & Free University of Brussels, (1993).

[10] DeLoach, S., "Multiagent Systems Engineering: A Methodology and Language for Designing Agent Systems", *Proceedings of Agent-Oriented Information Systems (AOIS'99)*, Seattle USA, (1999).

[11] Gómez-Sanz, J. and Pavón, J., "Agent Oriented Software Engineering with INGENIAS", *Proceedings of the 3rd Central and Eastern Europe Conference on Multiagent Systems*, Prague: Springer-Verlag, (2003).

[12] Huysman, M. and Wit, D., "Knowledge Sharing in Practice", *Kluwer Academic Publishers,* Dordrecht, (2000).

[13] Iglesias, C., Garijo, M., González, J., and Velasco, J. "A Methodological Proposal for Multiagent Systems

Development Extending CommonKads", *Proceedings of 10th KAW*, Banoe, Canada, (1996).

[14] Iglesias, C., Garijo, M., González, J.C., and Velasco, J., "Analysis and Design of MultiAgent Systems Using MAS-CommonKADS", *Intelligent Agents IV, M. P. Singh, A. Rao y M.J. Wooldridge, Springer Verlag. LNAI 1365*. (1998).

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

[16] Odell, J., Van-Dyke, P., and Bauer, B., "Extending UML for Agents", *Proceedings of the Agent-Oriented Information Systems Workshop at the 17th National Conference on Artificial Intelligence*, Austin USA, (2000).

[17] Pavón, J. and Gómez-Sanz, J., "Agent Oriented Software Engineering with INGENIAS", *Multi-Agent Systems and Applications III, 3rd International Central and Eastern European Conference on Multi-Agent Systems (CEEMAS 2003)*, LNCS 2691, Springer Verlag, (2003).

[18] Pavón, J. and Gómez-Sanz, J., *INGENIAS web site. http://grasia.fdi.ucm.es/ingenias/*, consulted at March 2006.

[19] Rodríguez, O., Martinez-García, Ana., Vizcaíno, A., Favela, J., and Piattini, M., "Identifying Knowledge Flows in Communities of Practice", in: Coakes, Elayne and Steve Cleark (Eds.), *Encyclopedia of Communities of Practice in Information and Knowledge Management*, IDEA Group Press, pp. 210-217, ISBN: 59140-556-4, (2005).

[20] Rus, I. and Lindvall, M., "Knowledge Management in Software Engineering", *IEEE Software*, Vol. 19, pp. 26-38, (2002).

[21] Skyrme, D., "Knowledge Management: Making Sense of an Oxymoron", *Management Insight*, Vol. 22, (2003).

[22] Soto, Juan P., Vizcaíno, A., and Piattini, M., "A Multiagent Architecture to Process Knowledge", *IADIS International Conference WWW/Internet*, Murcia Spain, (2006).

[23] Tacla, C. and Barthes, J-P., "A Multi-Agent Architecture for Knowledge Management System", *Second IEEE International Symposium on Advanced Distributed Computing Systems (ISADS)*, (2002).

[24] van-Elst, L., Dignum, V., and Abecker, A., "Agent-Mediated Knowledge Management", *International Simposium AMKM*, Stanford USA, Springer, (2003).

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

[26] Wooldridge, M., Jennings, N., and Kinny, D., "The Gaia Methodology for Agent-Oriented Analysis and Design", *Journal of Autonomous Agents and Multi-Agent Systems*, Vol. 3, no. 3, pp. 258-312, (2003).