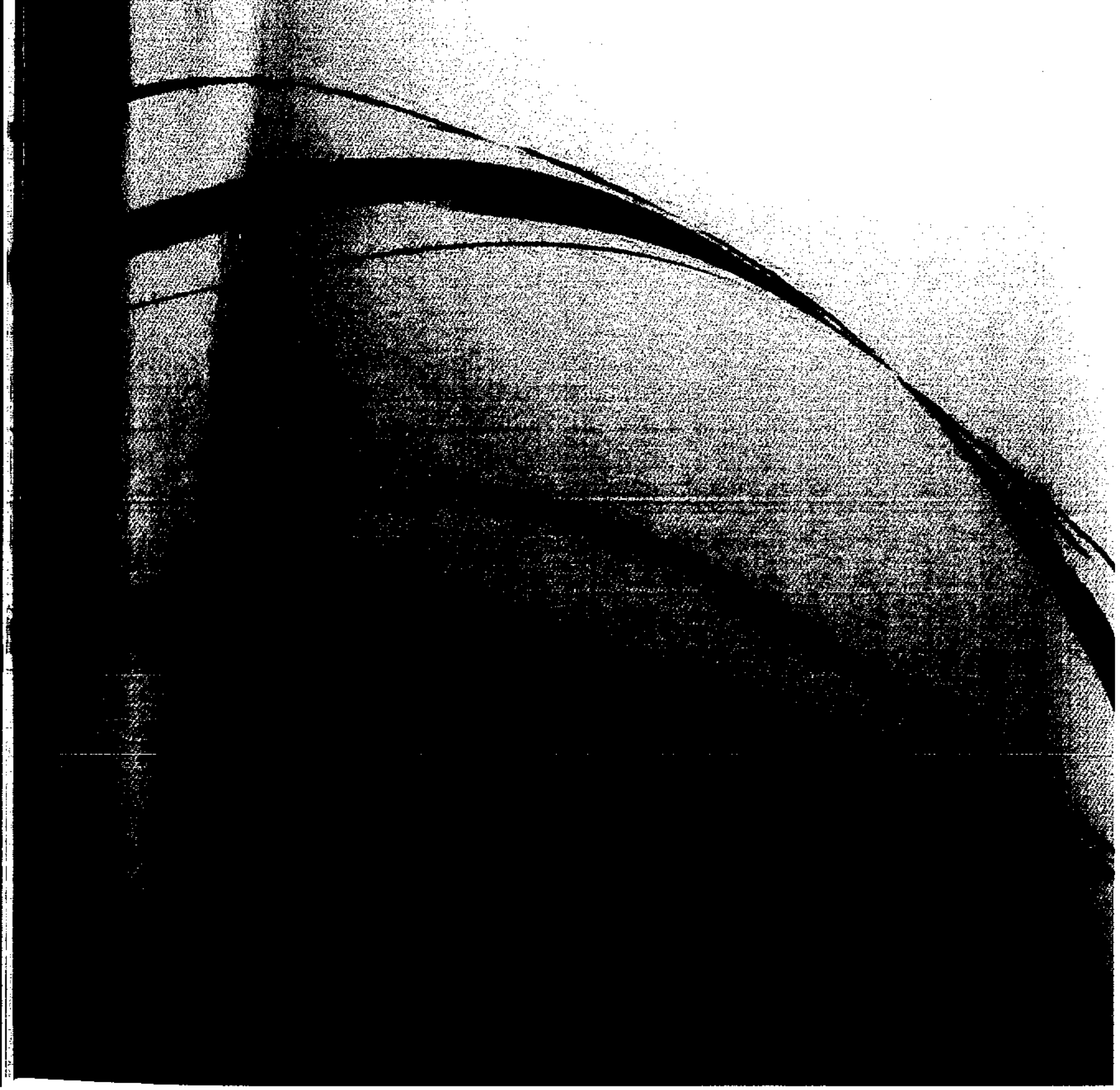
IBERAMIA'08

Ibero-American Conference on Artificial Intelligence

IBERAGENTS 2008 7th Ibero-American Workshop in Multi-Agent Systems

Luís Antunes / Luís Moniz / Juan Pavón

LISBOA /// October 14-17, 2008



IBERAGENTS 2008 7th Ibero-American Workshop in Multi-Agent Systems

Luís Antunes / Luís Moniz / Juan Pavón

7th Ibero-American Workshop in Multi-Agent Systems (Iberagents 2008)

Luis Antunes[1], Luis Moniz[1], Juan Pavón[2]

[1]LabMAg, University of Lisbon
Campo Grande, 1749-016 Lisbon, Portugal
xarax@di.fc.ul.pt, hal@di.fc.ul.pt
[2] Facultad de Informática, Universidad Complutense Madrid
Ciudad Universitaria s/n, 28040 Madrid, Spain
jpavon@fdi.ucm.es

Preface

Since 1994, the Iberagents workshop provides a discussion forum to the Ibero-American community on Multi-agent systems. The seventh Ibero-American Workshop in Multi-Agent Systems, has been organised as a part of the eleventh IBERAMIA conference, in Lisbon, on October 2008, in a close collaboration between LabMAg/University of Lisbon and Facultad de Informática/Universidad Complutense de Madrid. Iberagents 2008 is an international forum, with English as the official language for presenting and discussion the most recent and innovative work in the areas of multiagent systems and autonomous agents. Both academic researchers and industrial practitioners are welcome for discussion and exchange of ideas.

This time the discussion is structured along three working sessions, with the purpose of analysing different aspects that contribute to the application of agent technology, taking into account the degree of maturity of the discipline nowadays, which is represented by the contributions of the participants.

The first session, *Theories and Architectures*, starts by considering several issues on the fundamentals of agent technology. Some of them are quite general, looking at the relevance of specific characteristics of agents in the conception of distributed intelligent systems. In this sense, there are contributions related with the contribution of agents to improve software engineering practices. Other works provide insight on how agents can contribute to modelling real complex systems: Is it possible to study emergent behaviour of real systems by modelling as multi-agent systems?

The second session is devoted to *Applications*. The purpose is to analyse at which point there is more than prototypes by using agent technology. The agent paradigm, because of intentional and social characteristics, is appropriate to solve coordination problems involving heterogeneous parties. In this line, all the works present highly collaborative systems in different settings: video conferencing, airport management, and crisis management.

The third session, *Mechanisms*, provides insight on techniques associated to multiagent systems, this year looking especially at those issues that are derived from the social nature of multi-agent systems. Along those lines, the notion of adaptivity is very important, and the papers emphasise it through several different approaches: auctions, topologies, local interactions, reputation and trust.

The workshop programme has been completely integrated within IBERAMIA, covering a full day of the conference. Fifteen papers were submitted to the workshop, each one receiving three or four reviews, either from the Workshop's Programme Committee members or by their appointment. We would like to thank all Programme Committee members, as well as all of the reviewers, who took upon most of the burden in reviewing the papers enabling their accurate evaluation. Finally, we would like to thank all the authors who submitted their work to the workshop and enabled the success of Iberagents 2008.

Lisbon, September 2008
The organising committee:

Luis Antunes
Luis Moniz
Juan Pavón

The Program Committee was formed by:

Amilcar Cardoso (Universidade de Coimbra, Portugal)
Analia Amandi (UNICEN, Argentina)
Candelaria Sansores (Universidad del Caribe, Mexico)
Carlos Martinho (Universidade Técnica Lisboa, Portugal)
Christian Lemaitre (Autonomous Metropolitan University ,Mexico)
Cora Beatriz Excelente Toledo (LANIA, Mexico)
Emilio Corchado (Universidad de Burgos, Spain)
Eugénio Oliveira (Universidade Porto, Portugal)
Fernando Díaz (Universidad de Valladolid, Spain)
Francisco Garijo (Telefónica I+D, Spain)
Graça Gaspar (Universidade de Lisboa, Portugal)
Helder Coelho (Universidade de Lisboa, Portugal)
Javier Carbó (Universidad Carlos III de Madrid, Spain)
João Balsa (Universidade de Lisboa, Portugal)
João Leite (Universidade Nova de Lisboa, Portugal)
Jomi F. Hübner (University of Blumenau, Brazil)
Jordi Sabater-Mir (IIIA, Spain)
Jorge J. Gómez-Sanz (Universidad Complutense de Madrid, Spain)
Jorge Louçã (ISCTE, Portugal)
José Cascalho (Universidade Açores, Portugal)
José Machado (Universidade Minho, Portugal)
José R. Villar (Universidad de Oviedo, Spain)
Juan A. Botia (Universidad de Murcia, Spain)
Juan A. Rodríguez-Aguilar (IIIA, Spain)
Juan C. González Moreno (Universidad de Vigo, Spain)
Juan Manuel Corchado (Universidad de Salamanca, Spain)
Leonardo Garrido (Tecnológico de Monterrey, Mexico)
Luis Paulo Reis (Universidade Porto, Portugal)
Pablo Noriega (IIIA, Spain)
Paulo Cortez (Universidade Minho, Portugal)
Paulo Trigo (ISEL, Portugal)
Paulo Urbano (Universidade de Lisboa, Portugal)
Rafael Corchuelo (Universidad de Sevilla, Spain)
Rafael H. Bordini (University of Durham, UK)
Rosa Viccari (Universidade Federal do Rio Grande do Sul, Brasil)
Rubén Fuentes (Universidad Complutense de Madrid, Spain)
Vicente Julian (Universidad Politecnica de Valencia, Spain)
Viviane Torres da Silva (Universidad Complutense Madrid, Spain)
Wamberto Vasconcelos (University of Aberdeen,UK)

Table of Contents

A Two Layer Multi-agent Architecture to Support Communities of Practice: a Knowledge Management Perspective

Juan Pablo Soto, Aurora Vizcaíno, Javier Portillo-Rodríguez , Mario Piattini

University of Castilla – La Mancha
Alarcos Research Group – Institute of Information Technologies & Systems
Dep. of Information Technologies & Systems - Escuela Superior de Informática
Ciudad Real, Spain
JuanPablo.Soto@inf-cr.uclm.es, {Aurora.Vizcaino, Javier.Portillo, Mario.Piattini}@uclm.es

Abstract. This paper presents a two layer multi-agent architecture designed to support communities of practice in organizations which are concerned about knowledge sharing. The main goal of this proposal is, therefore to facilitate knowledge exchange in organizations whose employees are organized into communities. In order to test the architecture, a prototype with which to recommend documents in communities of practice is also described.

Keywords: Multi-agent Systems, Communities of Practice, Knowledge Management.

1 Introduction

In recent years, Knowledge Management Systems (KMS) have captured the attention of those communities which are worried about their employees' competitiveness. These kinds of organizations are constantly attempting to implement mechanisms which support and assist these employees in their daily tasks. However, most of these systems are focused on technological aspects [1]. As a result of this, these systems often contain worthless information or, on other occasions, their knowledge sources do not provide the confidence which is necessary for employees to reuse the information. Companies therefore create both social and technical networks in order to stimulate knowledge exchange [2]. An essential ingredient of sharing knowledge and information in organizations is that of Communities of Practice (CoPs), by which we mean groups of people who share a concern, a set of problems, or a passion about a topic, and whose knowledge and expertise in this area depends on their interacting on an ongoing basis [3]. The ability of a CoP to create a friendly environment for individuals with similar interests and problems, in which they can discuss a common subject matter, encourages the transfer and creation of new knowledge. CoPs are recognized as being efficient for knowledge transfer in general [4]. Many organizations report that such communities help to reduce problems caused by lack of communication and save time by "working smarter" [5]. For these reasons, we consider the modeling of CoPs into KMS to be an adequate method by which to encourage knowledge sharing.

In order to support CoPs we have designed a two layer multi-agent architecture in which agents attempt to assist individuals to share information, and which assists CoP members to work effectively together through a trustworthy network.

The layout of this paper is as follows: Section 2 describes the use of software agents in CoPs, along with a scenario to illustrate how agents collaborate to attain a common goal in a community. The design of a multi-agent architecture to support CoPs is presented in Section 3. Section 4 describes a prototype used to test the multi-agent architecture proposed. In Section 5, we describe related works and finally, in Section 6, conclusions and future work are summarized.

2 Agents as Community of Practice members

Agents have been proposed as solutions to the problem of information overload [6]. The agents' autonomous behavior is critical to the goal of this research since it can reduce the amount of work that employees have to carry out when using a KMS. A further 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 [6].

The basic idea behind this paradigm is that software agents perform tasks similar to those that a human would carry out. We have chosen the agent paradigm because it constitutes a natural metaphor for systems with purposeful interacting agents, and this abstraction is close to the human way of thinking about their own activities [7]. This foundation has led to an increasing interest in social aspects such as motivation, leadership, culture or trust [8].

Bearing this in mind, we have used software agents to represent members of a CoP. CoPs can be divided according to their objectives and scope into socially-oriented, commercially-oriented and professionally-oriented. We focus our research on the last one which consists of company employees who communicate and share information in order to support their professional tasks. In a CoP the agents can play the roles of the individuals in the organizations.

Thus, Figure 1 shows how we have modelled CoPs by using software agents. Two types of agents are therefore used: the *User Agent* and the *Manager Agent*.

The former is used to represent each person that may consult or introduce information in a knowledge base. Therefore, the User Agent can assume three types of behavior or roles which are similar to the tasks that a person might carry out in a community. The User Agent plays one role or another depending upon whether the person that it represents carries out one of the following actions:

- The person contributes new knowledge to the communities in which s/he is registered. In this case the User Agent plays the role of the *Provider*.
- The person uses knowledge previously stored in the community. The User Agent will therefore be considered as a *Consumer*.
- The person helps other users to achieve their goals by, for instance giving an evaluation of certain knowledge. In this case the role is that of a *Partner*. Figure 1

thus shows that in community 1 there are two User Agents playing the role of the partner, one User Agent playing the role of the Consumer and another playing that of a Provider.

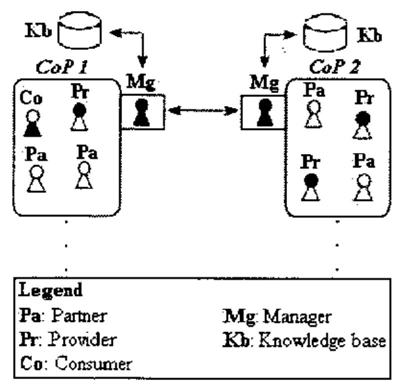


Fig. 1. Communities of Practice formed of agents.

The second type of agent within a community is called the Manager Agent (represented in black in Figure 1), which must manage and control its community. Both agents have been developed following the two layer architecture described in the following section.

3 A two layer multi-agent architecture

The goal of this work is to design a multi-agent architecture to support CoPs. The multi-agent architecture proposed is composed of two levels (see Figure 2): reactive and deliberative-social.

The reactive level is considered by other authors to be a typical level that a Multi-Agent System (MAS) must have [9]. A deliberative level is often also considered as a typical level but a social level is not frequently considered in an explicit way, despite the fact that these systems (MAS) are composed of several individuals, the interactions between them and the plans constructed by them. The social level is only considered in those systems that attempt to simulate social behavior. Since we wish to emulate human feelings such as trust and intuition when working in CoPs, we have added a social level that considers the social aspects of a community and which takes into account the opinions and behavior of each of the members of that community. Other previous works have also added a social level. For example, in [10] the author attempts to emulate human emotions such as fear, thirst or bravery, but in this case the author uses an architecture made up of three levels: reactive, deliberative and social. In our case the deliberative and social levels are not separate levels since we have realized that plans created in the deliberative level involve social interactions, We therefore consider that, in our case, it might be more efficient to define a level which is composed of two parts (deliberative-social level) rather than considering two separated levels.

Each of these levels is explained in greater detail in the following sub-sections.

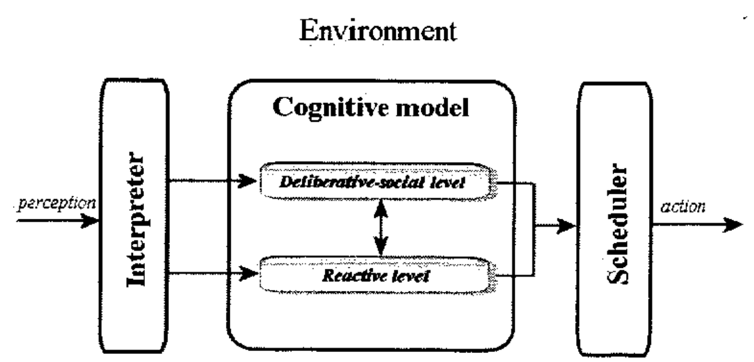


Fig. 2. Multi-agent architecture.

Two further important components of our architecture are the Interpreter and the Scheduler. The former is used to perceive the changes that take place. The Scheduler indicates how the actions should be executed.

3.1 Reactive level

This is the agent's capacity to perceive changes in its environment and respond to these changes at the precise moment at which they happen. It is in this level when an agent will execute another agent's request without any type of reasoning.

The components of the reactive level are (see Figure 3):

Internal model. This component stores the individual's features. These features will be consulted by other agents in order to discover more about the person represented by the User Agent. Therefore, this model stores the following information, which will be useful in calculating how trustworthy a knowledge source (in the case of CoPs, the members will be the knowledge sources since they contribute to the CoP with information) is, since one problem experienced by current CoPs is the lack of trust between members, as they are often geographically distributed and rarely experience face to face communication.

- *Expertise.* This information is an important factor since people often trust experts more than novice employees. The level of expertise that an individual has in a Cop could, for example, be calculated, from his/her CV or by considering the amount of time that a person has been working on a topic.
- *Position.* Employees often consider information that comes from a boss as being more reliable than that which comes from another employee in the same (or a lower) position as him/her [11]. However, this is not a universal truth and depends on the situation. For instance in a collaborative learning setting collaboration is more likely to occur between people of a similar status than between a boss and his/her employee or between a teacher and pupils [12]. Such different positions inevitably influence the way in which knowledge is acquired, diffused and eventually transformed within the CoP.
- *Previous experience.* Employees have greater trust in those sources from which they have previously obtained more "valuable information". Therefore, one factor which influences the increase or decrease in trust in a source is that of previous experience, and this factor can help us to detect trustworthy sources or knowledge.

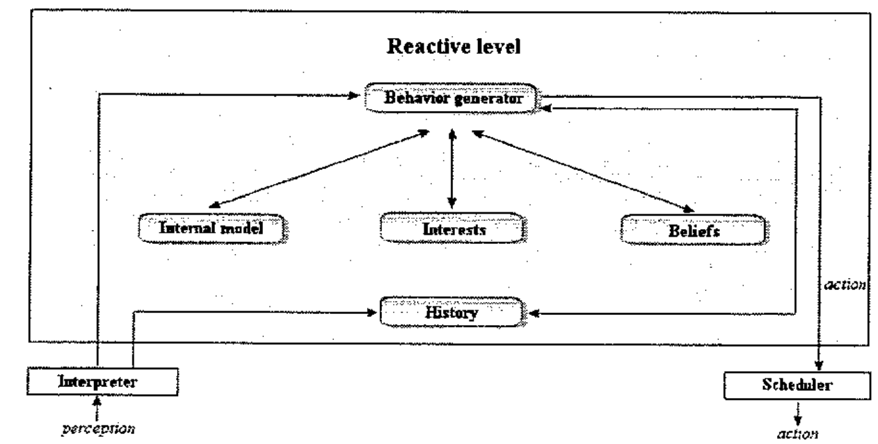


Fig. 3. Reactive level.

Interests. This component represents individual interests which represent the user's needs.

Beliefs. This module is composed of inherited beliefs and lessons learned from the agent itself. Inherited beliefs are the organization's beliefs that the agent receives such as the enterprise's organizational diagram or the organization's philosophy. Lessons learned are the lessons that the agent obtains while it interacts with the environment.

Behavior generator. This component is fundamental to our architecture. It is here that the actions to be executed by the agent are triggered. To do this, the behavior generator considers various information which comes from the internal model or the agent's interests and beliefs. This information is used by the behavior generator to generate an action, such as answering question about the level of expertise that the person who the agent represents has.

History. The history component stores the agent's interactions with its environment. This information represents the perceptions perceived by the interpreter and stored in the agent history. The history component also registers each of the actions executed by the agent in the environment. Finally, all the information stored by this component can be used to discover the knowledge sources which are most frequently consulted by or useful to the agents in the community.

3.2 Deliberative-Social level

At this level the agent has a type of behaviour which is oriented towards objectives, that is, it takes the initiative in order to plan its performance with the purpose of attaining its goals. At this level the agent would use the information that it receives from the environment and from its beliefs and intuitions to decide which is the best plan of action to follow in order to fulfill its objectives. This level has individual goals which pertain to the deliberative part, and social goals or cooperative goals which pertain to the social part.

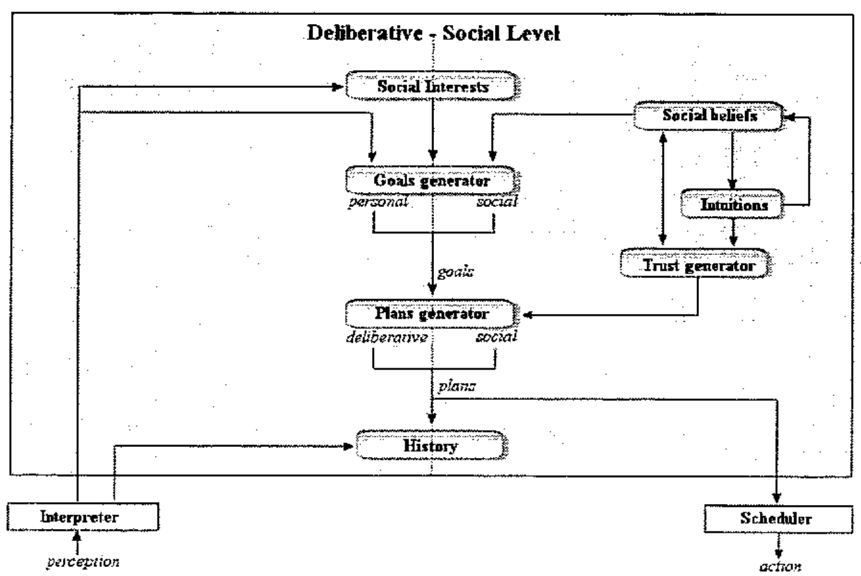


Fig. 4. Deliberative-social level.

The components of the deliberative-social level are (see Figure 4):

Goals generator. Depending on the state of the agent, this module must decide what the most important goal to be achieved is.

Social beliefs. This component represents a view that the agent has of the communities and their members. For instance, in this module there is information concerning community topics, in which areas other members are working, etc.

Intuitions. As we are modeling community members we have attempted to introduce factors into this architecture factors that influence people when they need to make decisions about whether or not to trust a knowledge source. One of these factors is intuition, which is a subjective factor since it is almost at the same level as a hunch and depends directly on each person's point of view. This concept is highly important when people do not have any previous experience. Other authors have called this issue "indirect reputation or prior-derived reputation" [13]. In human societies, each of us probably has different prior beliefs about the trustworthiness of strangers we meet. Sexual or racial discrimination might be a consequence of such prior belief [13]. We often trust more in people who have similar features to our own. For instance, when a person consults *Tripadvisor* [20] s/he often checks comments from people who are of the same age or have similar interests to him/her. In this research, intuition has therefore been modeled according to the similarity between agents' profiles: the greater the similarity between one agent and another, the greater the level of trust. The agents' profiles may change according to the community in which they are working.

Social interests. This module stores the interests of the community, such as identifying experts in the community, keeping the community knowledge updated, maintaining a friendly environment that provides users with the necessary confidence to share knowledge, etc.

Plans generator. This component is in charge of evaluating how a goal can be attained, and which plans are most appropriate to achieve this.

Trust generator. This module is in charge of generating a trust value for the knowledge sources with which an agent interacts in the community. To do this, the trust generator module considers the trust model explained in detail in [14] which considers the information obtained from the internal model and the agent's intuitions. This model permits us to calculate the trust level by using the following formula:

$$T_{sj} = w_e * E_j + w_p * P_j + w_i * I_{sj} + (\sum_{j=1}^{n} QC_{sj})/n \qquad (1)$$

where T_{sj} is the trust value of an Agent *j* in the eyes of another Agent *s* and E_j is the value of expertise which is calculated according to the degree of experience that the person upon whose behalf the agent acts has in a domain. P_j is the value assigned to a person's position. I_{sj} denotes the intuition value that Agent *s* has in the eyes of Agent *j*, and is calculated by comparing each of the user's profiles.

Previous experience should also be calculated. When an Agent *s* consults information from another Agent *j*, the Agent *s* should evaluate how useful that information has been. This value is called QC_{sj} (Quality of *j*'s Contribution in the opinion of *s*). To attain the average value of an agent's contributions, we calculate the sum of all the values assigned to these contributions and we divide it between their total. In the expression, n represents the total number of evaluated contributions.

Finally, w_e, w_p and w_i are weights with which the trust value can be adjusted according to the degree of knowledge that one agent has about another. Therefore, if an Agent *s* has had frequent interactions with another Agent *j*, then Agent *s* will give a zero to w_i since, in this case, previous experience is more important than intuition. The same may occur with w_e, w_p. The weights may, therefore, have the value of 0 or 1 depending on the previous experience that an agent has.

History. This component stores each agent's interactions with others agents in the communities. This component also stores the plans to be executed by the agent in its environment.

The following section describes a prototype based on our proposal, which was used to test our architecture.

4 Prototype

In order to test the architecture a prototype was implemented. The prototype provides the options of using community documents and updating trust values. For example, if a user selects a community topic and wishes to search for documents related to a topic, his/her user agent will contact other user agents which have documents concerning that topic, and the user agent will then calculate the trust value for each agent (by using Formula 1, as is explained in the previous section). This means that these agents are considered to be knowledge sources and the user agent needs to calculate which "knowledge source" is most trustworthy. Once these values have been calculated, the user agent only shows the user the documents which have come from the most trustworthy agents.

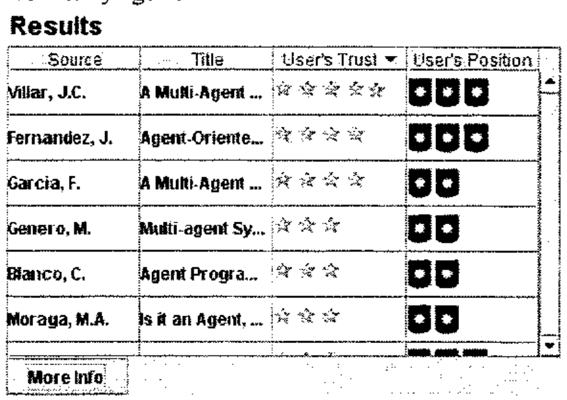


Fig. 5. Prototype interface.

Figure 5 shows the results of a search sorted by the trust values. The first documents on the list come from the most trustworthy knowledge sources (in this case the most trustworthy agents with the highest trust values). The trust level is represented by the number of stars. For instance, five stars indicate a high level confidence in that knowledge source and one star indicates the lowest level. The interface previously illustrated similarly shows the users' positions which indicate their level in the organization or community.

5 Related work

This research can be compared with others proposals which also use agents and the trust concept. For example, in [15] the authors present the Sporas model, a reputation mechanism for loosely connected online communities in which, among other features, new users start with a minimum reputation value, the reputation value of a user never falls below the reputation of a new user and users with very high reputation values experience much smaller rating changes after each update. The problem with this approach is that when somebody has a high reputation value it is difficult to change this reputation, or the system needs a high amount of interactions. A further approach of the Sporas authors is that of Histos which is a more personalized system than Sporas and is orientated towards highly connected online communities. In [16] the authors present another reputation model called REGRET in which the reputation values depend on time: the most recent rates are more important than previous rates. In [17] the authors present the AFRAS model, which is based on Sporas. The authors present a complex computing reputation mechanism which handles reputation as a fuzzy set, while decision making is inspired in a cognitive human-like approach. In [18] the authors present a trust and reputation model that considers trust and reputation as emergent properties of direct interactions between agents, based on multiple interactions between two parties. In this model, trust is a belief that an agent has about the performance of the other party to solve a given task, according to its own knowledge. Another interesting work is that of Barber and Kim which presents a multi-agent belief revision algorithm based on belief networks [19]. In their model, the agent is able to evaluate incoming information, to generate a consistent knowledge base, and to avoid fraudulent information from unreliable or deceptive information sources or agents. This work has a similar goal to ours. However, the means of attaining it are different. In Barber and Kim's case they define reputation as a probability measure, since the information source is assigned a reputation value of between 0 and 1. Moreover, every time a source sends knowledge that source should indicate the certainty factor that the source has of that knowledge. In our case, the focus is very different since it is the receiver who evaluates the relevance of a piece of knowledge rather than the provider as in Barber and Kim's proposal.

6 Conclusions and future work

CoPs are means of knowledge sharing. However, if it is to be reused, knowledge should be valuable. Otherwise CoPs members often prefer to ignore the knowledge that a community has. In order to encourage the reuse of knowledge in CoPs, in this work we propose a two-layer architecture in which agents represent CoPs members and use a trust model to evaluate how trustworthy a knowledge source is. Other related works (mentioned in the previous section) do not describe any kind of architecture. One advantage of this approach is that this architecture is specially designed to be used in CoPs, and has consequently taken into account factors such as expertise, position, intuition and previous experience, which influence whether or not people trust a knowledge source. This architecture works when there is no previous experience, and takes other factors, such as expertise, intuition and position into account, and this is a very important advantage over other works which need a large amount of experience. In our case, the architecture is capable of calculating a preliminary trust value of a knowledge source without using the previous experience factor. A further advantage of our architecture is that it considers social aspects which are implemented in the deliberative social layer. In fact, the main feature of this architecture is contained in the Deliberative-Social level, which arose after designing a previous version of the architecture with three levels (reactive, deliberative and social). However, we realised that in our domain, all deliberative goals implied social behaviour. As a result of this, the social and deliberative levels were joined into one level, and this is one of the differences between our architecture and others which use three levels. Another contribution of our architecture is that it models the intuition concept which, owing to its subjective character, is not often considered in multi-agents systems.

As future work, we are planning to add new functions to the prototype, such as expert detection and recognition of fraudulent members who contribute with worthless knowledge.

Acknowledgments. This work is partially supported by the MELISA project (PAC08-0142-3315), Junta de Comunidades de Castilla-La Mancha, Consejería de Educación y Ciencia, in Spain, partially supported by MECENAS (PBI06-0024) and ESFINGE project (TIN2006-15175-C05-05) Ministerio de Educación y Ciencia (Dirección General de Investigación)/Fondos Europeos de Desarrollo Regional (FEDER) in Spain and CONACYT (México) under grant of the scholarship 206147 provided to the first author.

References

1. Hahn, J. and Subramani, M.: A Framework of Knowledge Management Systems: Issues and Challenges for Theory and Practice. In Proceedings of the Twenty First International Conference on Information Systems, pp. 302-312, (2000)
2. Vizcaíno, A., Soto, J.P., Portillo-Rodríguez, J., Piattini, M.: Developing Knowledge Management Systems from a Knowledge-Based and Multi-Agent Approach. In International Journal of Knowledge Management (IJKM), Vol. 3(4), pp. 67-82, (2007)
3. Wenger, E.: Communities of Practice: Learning Meaning, and Identity. Cambridge University Press, (1998)
4. Gammelgaard, J. and Ritter, T.: Virtual Communities of Practice: A Mechanism for Efficient Knowledge Retrieval in MNCs. In International Journal of Knowledge Management (IJKM), Vol. 4(2), pp. 46-51, (2008)
5. Wenger, E., McDermott, R. and Snyder, W.: Cultivating Communities of Practice. Harvard Business School Press (2002)
6. Maes, P.: Agents that Reduce Work and Information Overload. In Communications of the ACM, Vol. 37(7), pp. 31-40, (1994)
7. Wooldridge, M. and Ciancarini, P.: Agent-Oriented Software Engineering: The State of the Art. In Wooldridge M., Ciancarini, P. (Eds.), Agent Oriented Software Engineering. Springer Verlag, LNAI 1975, (2001)

8. Fuentes, R., Gómez-Sanz, J. and Pavón, J.: A Social Framework for Multi-agent Systems Validation and Verification. In Wang, S. et al (Eds.) ER Workshop 2004, Springer Verlag, LNCS 3289, pp. 458-469 (2004)
9. Ushida, H., Hirayama, Y. and Nakajima, H.: Emotion Model for Life like Agent and its Evaluation. In Proceedings of the Fifteenth National Conference on Artificial Intelligence and Tenth Innovative Applications of Artificial Intelligence Conference (AAAI'98 / IAAI'98). Madison, Wisconsin, USA, (1998)
10. Imbert, R., and de Antonio, A.: When emotion does not mean loss of control. Lecture Notes in Computer Science, T. Panayiotopoulos, J. Gratch, R. Aylett, D. Ballin, P. Olivier, and T. Rist (Eds.), Springer-Verlag, London UK, pp. 152-165, (2005)
11. Wasserman, S. and Glaskiewics, J.: Advances in Social Networks Analysis. Sage Publications, (1994)
12. Dillenbourg, P.: Introduction: What Do You Mean By "Collaborative Learning"?. In Collaborative Learning. Cognitive and Computational Approaches. Dillenbourg (Ed.). Elsevier Science, (1999)
13. Mui, L., Halberstadt, A. and Mohtashemi, M.: Notions of Reputation in Multi-Agents Systems: A Review. In International Conference on Autonomous Agents and Multi-Agents Systems (AAMAS'02), pp. 280-287, (2002)
14. Soto, J.P., Vizcaíno, A., Portillo-Rodríguez, J., Rodríguez-Elias, O-M., Piattini, M.: A Prototype to Recommend Trustworthy Knowledge in Communities of Practice. In 3rd International Conference on Software and Data Technologies (ICSOFT). Porto Portugal, pp. 321-326, (2008)
15. Zacharia, G., Moukas, A. and Maes, P.: Collaborative Reputation Mechanisms in Electronic Marketplaces. In 32nd Annual Hawaii International Conference on System Science (HICSS-32), Vol. 8, pp. 8026, (1999)
16. Sabater, J., Sierra, C.: Social REGRET, a Reputation Model based on social relations. Proceedings of the Fifth International Conference on Autonomous Agents, Vol. 3(1), pp. 44-56, (2002)
17. Carbo, J., Molina, M. and Davila, J.: Trust Management through Fuzzy Reputation. In International Journal of Cooperative Information Systems, Vol. 12(1), pp. 135-155, (2003)
18. Caballero, A., Botía, J. and Skarmeta, A.: A New Model for Trust and Reputation Management with an Ontology Based Approach for Similarity Between Tasks. In: Fischer, K., Timm, I. J., André, E., Zhong, N. (eds.) MATES. LNCS 4196, pp. 172-183 (2006)
19. Barber, K. and Kim, J.: Belief Revision Process Based on Trust: Simulation Experiments. In 4th Workshop on Deception, Fraud and Trust in Agent Societies. Montreal Canada, (2004)
20. Tripadvisor, URL: *http://www.tripadvisor.com* (2007)