# INTELIGENCIA ARTIFICIAL

HOME    ABOUT    LOG IN    REGISTER    SEARCH    CURRENT    ARCHIVES

ANNOUNCEMENTS

Home > **INTELIGENCIA ARTIFICIAL**

## INTELIGENCIA ARTIFICIAL

## Table of Contents

### JOURNAL CONTENT

Search

All

Search

Browse
- By Issue
- By Author
- By Title
- Other Journals

### USER

Username

Password

☐ Remember me

Log In

ATOM 1.0
RSS 2.0
RSS 1.0

### LANGUAGE

English

### INFORMATION

- For Readers
- For Authors
- For Librarians

# INTELIGENCIA ARTIFICIAL

# A Two-Layer Multi-agent Architecture to Facilitate Knowledge Sharing within Communities of Practice

Juan Pablo Soto[1], Aurora Vizcano[1], Javier Portillo[1], Mario Piattini[1], Oliver Kusche[2]

[1]Universidad de Castilla-La Mancha
Alarcos Research Group - Institute of Information Technologies and Systems
Dep. of Information Technologies and Systems - Escuela Superior de Informtica
Ciudad Real, Spain
juanpablo.soto@inf-cr.uclm.es, [aurora.vizcaino, javier.portillo, mario.piattini]@uclm.es

[2]Institute for Applied Computer Science (IAI)
Forschungszentrum Karlsruhe
Karlsruhe, Germany
oliver.kusche@iai.fzk.de

**Abstract** This paper presents a two-layer multi-agent architecture designed to support communities of practice in organizations which are concerned with knowledge sharing. The main goal of this proposal is 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 organizations which are worried about their employees' competitiveness. These organizations are constantly attempting to put in place mechanisms which support and assist these employees in their daily tasks. However, most of these systems are focused on technological aspects [10]. As a result, these systems often contain worthless information or, in other cases, their knowledge sources do not provide the confidence needed by employees for the reuse of such information. Companies, therefore, create both social and technical networks in order to stimulate knowledge exchange [19]. 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 interaction on an ongoing basis [22]. 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 [9]. Many organizations report that such communities help to reduce problems caused by lack of communication and save time by "working smarter" [23]. For these reasons, we consider that the modeling of CoPs into KMS is a suitable 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 proposed multi-agent architecture. In Section 5, we describe related works and finally, in Section 6, conclusions and future work are summarized

## 2    Agents as Communities of Practice Members

Agents have been proposed as solutions to the problem of information overload [12]. 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. In addition, intelligent agents' specific characteristics turn them into promising candidates in providing a KMS solution [13]. Moreover, software agent technology can: monitor and coordinate events, meetings and disseminate information [2], automate of complex industrial processes [7], build and maintain organizational memories [1]. Another important issue is that agents can learn from their own experience. Consequently, agent systems are expected to become more efficient over time since the agents learn from their previous mistakes and successes [12].
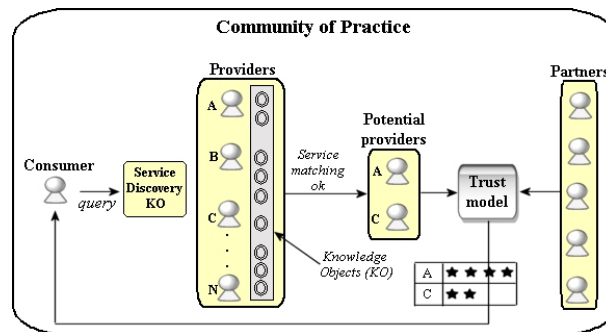


Figure 1: Community of Practice formed of agents.

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 [24]. 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 CoP by using software agents. The User Agent is used to represent each person that may consult or introduce information into 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. In the community shown in Figure 1, there are *five* User Agents playing the role of the partner, *one* User Agent playing the role of the Consumer and $n$ playing that of a *Provider*.

The 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 and deliberative levels are considered by other authors to be a typical level that a Multi-Agent System (MAS) must have [18]. On the other hand, the social level is only considered in those systems that attempt to simulate social behaviour. Since we wish to emulate human aspects 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 member on the community. Other previous works have also added a social level. For example, in [11] the authors attempt 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 separate levels. Each of these levels is explained in greater detail in the following sub-sections.
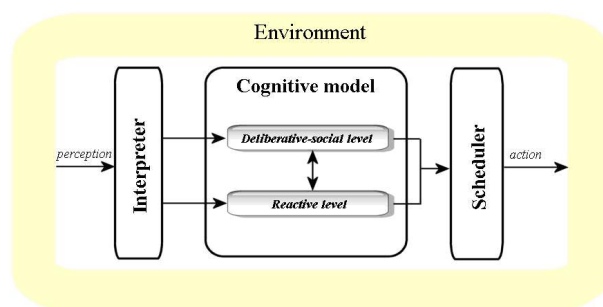


Figure 2: Multi-agent architecture.

## 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, for instance 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. In the case of CoPs, the members will be the knowledge sources since they contribute to the CoP with information. 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. Therefore, the model stores the following information, which will be useful to calculate the trustworthiness of a knowledge source:

- *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.
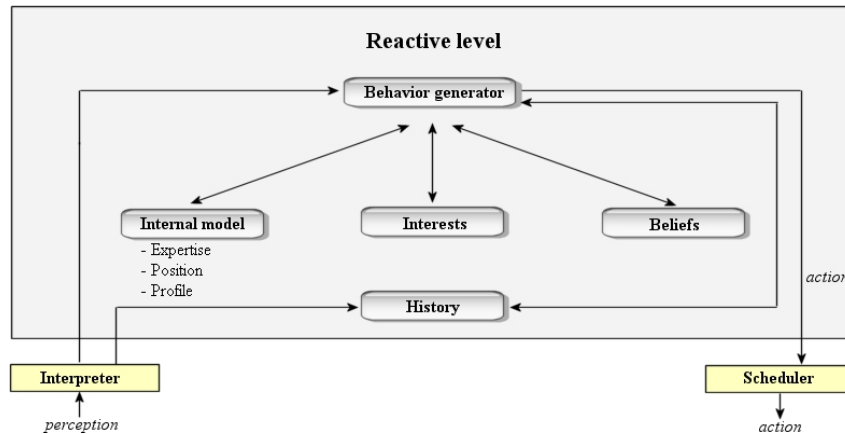
Figure 3: Reactive level

- *Position.* Employees often consider information that comes from a superior as being more reliable than that which comes from another employee in the same (or a lower) position as him/her [21]. 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 superior and his/her employee or between a teacher and pupils [6]. Such different positions inevitably influence the way in which knowledge is acquired, diffused and eventually transformed within the CoP.

- *Profile* This part is included in the internal model to describe the profile of the person on whose behalf the agent is acting. Therefore, a person's preferences are stored here.

**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 where the actions to be executed by the agent are triggered. To do this, the generator of behavior considers various information which comes from the internal model, or from the interests and beliefs of the agent. This information is used by the generator of behavior to generate an action, such as answering questions about the level of expertise that has the person who the agent represents.

**History**. The history component stores the agent's interactions with its environment. This information represents the perceptions achieved 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.

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.
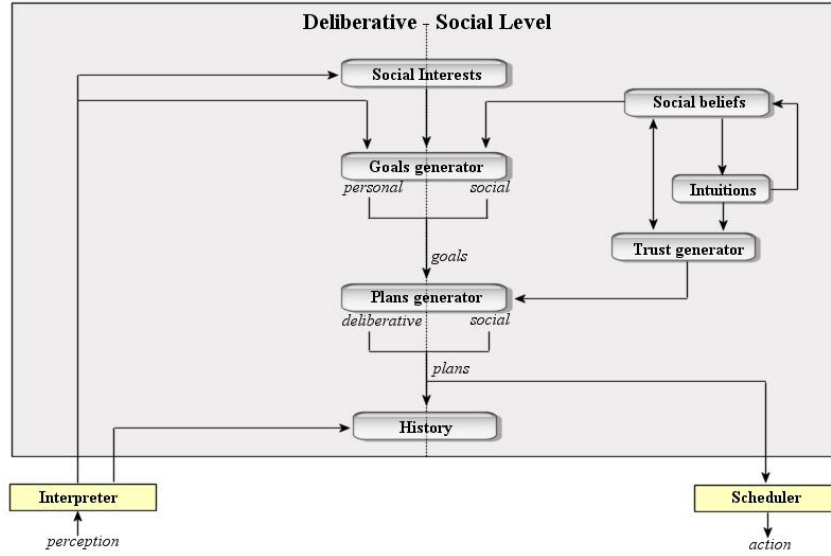
Figure 4: Deliberative-Social level

**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 depends on the individual person. 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" [14]. 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 [14]. We often trust more in people who have similar features to our own. For instance, when a person consults a community for rating products or services such as Tripadvisor [17], 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 uses the trust model explained in detail in [20][16] which considers the information obtained from the internal model and the agent's intuitions. This model allows us to calculate the trust level by using the following formula:

$$T_{sj} = w_e * E_j + w_p * P_j + w_i * I_{sj} + \frac{1}{n}\sum_{j=1}^{n} QC_{sj} \tag{1}$$

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 $Q_{sj}$

(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 other 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 instance, 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.
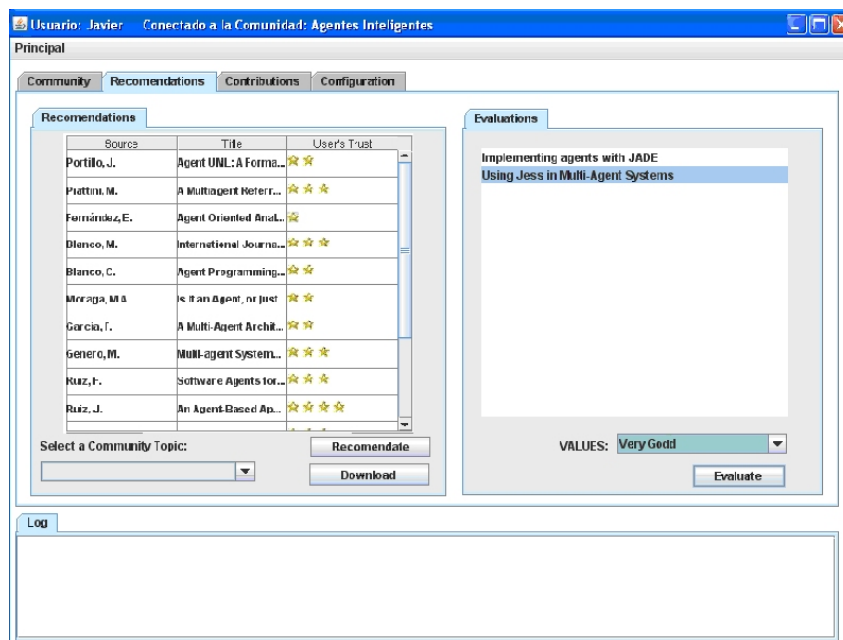


Figure 5: Prototype interface

Figure 5 shows the results of a search sorted by the trust values (on the left) and (on the right) there is a list of documents that the user has used and which should be evaluated. 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.

# 5   Related work

This research can be compared with others proposals which also use agents and the trust concept. For instance, in [25] 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 [15] 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 [5] 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 [4] 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 [3]. 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 CoP members might 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, due 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.

# Acknowledgements

# References

[1] A. Abecker, A. Bernardi, and L. van Elst. Agent technology for distributed organizational memories: The frodo project. *Proceedings of 5th International Conference on Enterprise Information Systems (ICEIS)*, 2:3–10, 2003.

[2] S. Balasubramanian, R. Brennan, and D. Norrie. An architecture for metamorphic control of holonic manufacturing systems. *Computers in Industry*, 46:13–31, 2001. doi: 10.1016/S0166-3615(01)00101-4

[3] K. Barber and J. Kim. Belief revision process based on trust: Simulation experiments. *4th Workshop on Deception, Fraud and Trust in Agent Societies*, pages 1–12, 2004.

[4] A. Caballero, J. Botía, and A. Skarmeta. 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*, pages 172–183, 2006.

[5] J. Carbo, M. Molina, and J. Davila. Trust management through fuzzy reputation. *International Journal of Cooperative Information Systems*, 12:135–155, 2003. doi: 10.1142/S0218843003000681

[6] P. Dillenbourg. Introduction: What do you mean by 'collaborative learning'? *Collaborative Learning Cognitive and Computational Approaches. Dillenbourg (Ed.), Elsevier Science*, 1999.

[7] J.C. Encinas, J. de Las Morenas, A. García, and J.M. Pastor. Rfid-imsii methodology for the management of a real factory simulation connected with a physical platform. *IMS'08: 9th IFAC Workshop on Intelligent Manufacturing Systems*, pages 151–156, 2008.

[8] R. Fuentes, J. Gómez-Sanz, and J. Pavón. A social framework for multi-agent systems validation and verification. *In Wang, S. et al (Eds.) ER Workshop, Springer Verlag, LNCS 3289*, pages 458–469, 2004.

[9] J. Gammelgaard and T. Ritter. Virtual communities of practice: A mechanism for efficient knowledge retrieval in mncs. *International Journal of Knowledge Management(IJKM)*, 4:46–51, 2008.

[10] J. Hahn and M. Subramani. A framework of knowledge management systems: Issues and challenges for theory and practice. *Proceedings of the Twenty First International Conference on Information Systems*, pages 302–312, 2000.

[11] R. Imbert and A. de Antonio. When emotion does not mean loss of control. *In T. Panayiotopoulos, J. Gratch, R. Aylett, D. Ballin, P. Olivier, and T. Rist, (Eds.), LNCS, Springer-Verlag*, pages 152–165, 2005.

[12] P. Maes. Agents that reduce work and information overload. *Communications of the ACM*, 37:31–40, 1994.

[13] S. Mercer and S. Greenwood. A multi-agent architecture for knowledge sharing. *Proceedings of the Sixteenth European Meeting on Cybernetics and Systems Research*, 2001.

[14] L. Mui, A. Halberstadt, and M. Mohtashemi. Notions of reputation in multi-agents systems: A review. *International Conference on Autonomous Agents and Multi-Agents Systems (AAMAS)*, pages 280–287, 2002.

[15] J. Sabater and C. Sierra. Social regret, a reputation model based on social relations. *Proceedings of the Fifth International Conference on Autonomous Agents*, 3:44–56, 2002.

[16] J.P. Soto, A. Vizcaíno, J. Portillo-Rodríguez, O-M. Rodríguez-Elias, and M. Piattini. A prototype to recommend trustworthy knowledge in communities of practice. *3rd International Conference on Software and Data Technologies (ICSOFT)*, pages 321–326, 2008.

[17] TripAdvisor. 2008. `http://www.tripadvisor.com`.

[18] H. Ushida, Y. Hirayama, and H. Nakajima. Emotion model for life like agent and its evaluation. *Proceedings of the Fifteenth National Conference on Artificial Intelligence and Tenth Innovative Applications of Artificial Intelligence Conference (AAAI / IAAI)*, pages 8–37, 1998.

[19] A. Vizcaíno, J.P. Soto, J. Portillo-Rodríguez, and M. Piattini. Developing knowledge management systems from a knowledge-based and multi-agent approach. *International Journal of Knowledge Management (IJKM)*, 3:67–82, 2007.

[20] A. Vizcaíno, J. Portillo-Rodríguez, J.P. Soto, and M. Piattini. Encouraging the reuse of knowledge in communities of practice by using a trust model. *International Conference on Information, Process, and Knowledge Management (eKNOW), IEEE Computer Society*, pages 28–33, 2009.

[21] S. Wasserman and J. Glaskiewics. Advances in social networks analysis. *Sage Publications*, 1994.

[22] E. Wenger. Communities of practice: Learning meaning, and identity. *Cambridge University Press, Cambridge U.K.*, 1998.

[23] E. Wenger, R. McDermott, and W. Snyder. Cultivating communities of practice: A guide to managing knowledge. *Harvard Business School Press*, 2002.

[24] M. Wooldridge and P. Ciancarini. Agent-oriented software engineering: The state of the art. *In Wooldridge M., Ciancarini, P. (Eds.), Agent Oriented Software Engineering. Springer Verlag, LNAI 1975*, 2001.

[25] G. Zacharia, A. Moukas, and P. Maes. Collaborative reputation mechanisms in electronic marketplaces. *32nd Annual Hawaii International Conference on System Science (HICSS)*, 1999.