ISSN 1557-3958

International Journal of Cognitive Informatics and Natural Intelligence

Vol. 3, No. 1, 2009

Vol. 3, No. 1
January - March 2009

An official publication of the Information Resources Management Association

Special Issue: Autonomous Agent Systems (AAS's)

INTERNATIONAL JOURNAL OF Cognitive Informatics and Natural Intelligence

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

The *International Journal of Cognitive Informatics and Natural Intelligence* (ISSN: 1557-3958 e-ISSN: 1557-3966) is published quarterly by IGI Global, 701 E. Chocolate Avenue, Suite 200, Hershey, PA 17033-1240, USA, www.igi-global.com. For pricing, please visit www.igi-global.com/journals. Subscription to this journal is $90 individual, $375 online, and $425 for institutions. **Advertising information:** IGI Global, 701 E. Chocolate Avenue, Suite 200, Hershey, PA 17033-1240, USA,Tel: (717) 533-8845, Fax: (717) 533-8661. **Postmaster:** Send all address changes to IGI Global.

JCINI is currently listed or indexed in: CSA Illumina; DBLP; Google Scholar; Index of Information Systems Journals; PsycINFO

IGI Publishing

Hershey • New York

INTERNATIONAL JOURNAL OF COGNITIVE INFORMATICS AND NATURAL INTELLIGENCE

January-March 2009, Vol. 3, No. 1

Table of Contents

International Journal of Cognitive Informatics and Natural Intelligence

Guidelines for Manuscript Submissions

Mission: Cognitive Informatics (CI) is a cutting-edge and multidisciplinary research area that tackles the fundamental problems shared by modern informatics, computation, software engineering, AI, cybernetics, cognitive science, neuropsychology, medical science, systems science, philosophy, linguistics, economics, management science, and life sciences. CI is the transdisciplinary study into the internal information processing mechanisms and processes of the Natural Intelligence (NI) – human brains and minds – and their engineering applications in computing, ICT, and healthcare industries.
The development and the cross fertilization between the aforementioned science and engineering disciplines have led to a whole range of extremely interesting new research areas known as CI and NI. The International Journal of Cognitive Informatics and Natural Intelligence (IJCINI) encourages submissions that transcends disciplinary boundaries, and is devoted to rapid publication of high quality articles. The themes of IJCINI are Natural Intelligence, Autonomic Computing, and Neuroinformatics. IJCINI is expected to provide the first forum and platform in the world for researchers, practitioners, and graduate students to investigate cognitive mechanisms and processes of human information processing, and to stimulate the transdisciplinary effort on cognitive informatics and natural intelligent research and engineering applications.

Scope: Original articles are invited from multidisciplinary perspectives on subject areas including, but not limited to, the following:

Natural Intelligence (NI)	Autonomic Computing (AC)	Neural Informatics (NeI)
Informatics models of the brain	Imperative vs. autonomous computing	Neuroscience foundations of information processing
Cognitive processes of the brain	Reasoning and inferences	Cognitive models of the brain
Internal information processing mechanisms	Cognitive informatics foundations of AC	Functional modes of the brain
Theories of natural intelligence	Memory models	Neural models of memory
Intelligent foundations of computing	Informatics foundations of software engineering	Neural networks
Descriptive mathematics for NI	Fuzzy logic	Neural computation
Abstraction and means	Knowledge engineering	Cognitive linguistics
Ergonomics	Pattern recognition	Neuropsychology
Informatics laws of software	Agent technologies	Bioinformatics
Knowledge representation	Artificial intelligence	Biosignal processing
Models of knowledge and skills	Software agent systems	Cognitive signal processing
Language acquisition	Decision theories	Gene analysis
Cognitive complexity of software	Problem solving	Gene expression
Distributed intelligence	Machine learning	Neural signal interpretation
Computational intelligence	Intelligent Internet	Visual information representation
Emotions/motivations/attitudes	Web contents cognition	Visual information interpretation
Perception and consciousness	Nature of software	Sensational cognitive processes
Hybrid (AI/NI) intelligence	Quantum computing	Human factors in systems

Originality: Prospective authors should note that only original and previously unpublished manuscripts will be considered. Furthermore, simultaneous submissions are not acceptable. Submission of a manuscript is interpreted as a statement of certification that no part of the manuscript is copyrighted by any other publisher nor is under review by any other formal publication. It is the primary responsibility of the author to obtain proper permission for the use of any copyrighted materials in the manuscript prior to the submission.

Style: Submitted manuscripts must be written in the APA (American Psychological Association) editorial style. References should relate only to material cited within the manuscript and be listed in alphabetical order, including the author's name, complete title of the cited work, title of the source, volume, issue, year of publication, and pages cited. Please do not include any abbreviations. See the following examples:

Example 1: Single author periodical publication.
Smith, A. J. (1988). Microcomputers and organizations. *Microcomputer Ideology Review, 16*(2), 1-5.

Example 2: Multiple authors periodical publication.
Smith, A. J., & Brown, C. J. (1988). Organizations and microcomputer processing. *Microcomputer Source, 10*(4), 77-88.

Example 3: Books.
Smith, A. J. (1989). *Microcomputer booklet.* New York: J.J. Press.

State author's name and year of publication where you use the source in the text. See the following examples:

Example 1: In most organizations, microcomputer resources are considered to be a major resource (Brown, 1988; Smith, 1987).

Example 2: Brown (1989) states that the value of microcomputers is recognized by most organizations.

Direct quotations of another author's work should be followed by the author's name, date of publication, and the page(s) on which the quotation appears in the original text.

Example 1: Brown (1989) states that "the value of microcomputers is realized by most organizations" (p. 45).

Example 2: In most organizations, "microcomputer resources are considered to be a major organization asset" (Smith, 1988, pp. 35-36) and must be carefully monitored by the senior management.

For more information please consult the APA manual or review previous issues of the Journal of Organizational and End User Computing.

Review Process of IJCINI: To ensure the high quality of published material, IJCINI utilizes a group of experts to review submitted manuscripts. Each submission is reviewed on a blind basis by at least three members of the International Editorial Board of the journal. A revised manuscript should be returned to the Editor in Chief with a statement that describes actions taken in the revision corresponding to the main review comments of referees. Revised manuscripts may be reviewed again by the original review panel with the addition of one new reviewer. Return of a manuscript to the author(s) for revision does not guarantee acceptance of the manuscript for publication. The final decision will be based upon the comments of the reviewers and associate editors, upon their final review of the revised manuscript.

Copyright: Authors are asked to sign a warranty and copyright agreement upon acceptance of their manuscript, before the manuscript can be published. All copyrights, including translation of the published material into other languages are reserved by the publisher, IGI Global. Upon transfer of the copyright to the publisher, no part of the manuscript may be reproduced in any form without written permission of the publisher, except for noncommercial, educational use such as for classroom teaching purposes.

Correspondance: All inquiries and submissions should be sent to:

Editor-in-Chief, Prof. Yingxu Wang
International Journal of Cognitive Informatics and Natural Intelligence (IJCINI)
The International Center for Cognitive Informatics (ICfCI)
Dept. of Electrical and Computer Engineering
Schulich School of Engineering
University of Calgary
2500 University Dr, NW
Calgary, AB, Canada T2N 1N4
Tel: (403) 220 6141
Fax: (403) 282 6855
Email: yingxu@ucalgary.ca
URL: http://www.enel.ucalgary.ca/IJCINI/

Manuscripts should be submitted as an email attachment. An acknowledgment email regarding the receipt of the manuscript will be promptly sent. The review process will take approximately 8-16 weeks, and the author will be notified concerning the possibility of publication of the manuscript as soon as the review process is completed. All correspondence will be directed to the first author of multi-authored manuscripts. It is the responsibility of the first author to communicate with the other co-author(s).

Authors of accepted manuscripts will be asked to provide a final copy of their manuscript in Word format stored on a CD, accompanied by a hard copy of the manuscript and the signed copy of the Warranty and Copyright Agreement. The accepted manuscript will be edited by the Journal copy editor for format and style.

An Agent System to Manage Knowledge in CoPs

Juan Pablo Soto, University of Castilla - La Mancha, Spain

Aurora Vizcaíno, University of Castilla - La Mancha, Spain

Javier Portillo-Rodríguez, University of Castilla - La Mancha, Spain

Mario Piattini, University of Castilla - La Mancha, Spain

ABSTRACT

This paper proposes a multi-agent architecture and a trust model with which to foster the reuse of information in organizations which use knowledge bases or knowledge management systems. The architecture and the model have been designed with the goal of giving support to communities of practices which are a means of sharing knowledge. However, members of these communities are currently often geographically distributed, and less trust therefore exists among members than in traditional co-localizated communities of practice. This situation has led us to propose our trust model, which can be used to calculate what piece of knowledge is more trustworthy. The architecture's artificial agents will use this model to recommend the most appropriate knowledge to the community's members.

Keywords: *agents; communities of practice; knowledge utilization*

INTRODUCTION

The need to support knowledge processes in organizations has always existed. However, its importance has definitely increased in the last few years. Recently, the concept of knowledge management suggests a paradox since compared with traditional production factors knowledge is so complex, scattered and hidden that it is rather complicated to manage it.

On the other hand, traditional Knowledge Management Systems (KMS) have received certain criticism as they are often implanted in companies overloading employees with extra work; for instance, employees have to introduce information into the KMS and worry about updating this information. As a result of this, these systems are sometimes not greatly used by the employees since the knowledge that these systems have is often not valuable or on other occasions the knowledge sources do not provide the confidence necessary for employees to reuse the information. Reusing information and not reinventing the wheel are frequently heard arguments. For this purpose, companies create both social and technical networks in order to stimulate knowledge exchange. An essential ingredient of knowledge sharing information in organizations is that of "community of practice", by which we mean groups of people with a common interest where each member contributes knowledge about a common domain (Wenger, 1998). The ability of a community of practice 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. Many companies report that such communities help reduce problems caused by lack of communication, and save time by "working smarter" (Wenger et al, 2002). In addition, communities of practice provide their members with the confidence to share information with each other. Moreover, individuals are frequently more likely to use knowledge built by their community team members than that created by members outside their group (Desouza et al, 2006). For these reasons, we consider the modelling of communities of practice into KMS as an adequate method by which to provide these systems with a certain degree of control to measure the confidence and quality of information provided by each member of the community.

In order to carry this out, we have designed a multi-agent architecture in which agents try to emulate human behaviour in communities of practice with the goal of fostering the use and exchange of information where intelligent agents suggest "trustworthy knowledge" to the employees and foster the knowledge flow between them.

The remainder of this work is organized as follows. The next section focuses on community of practice then in section 3 two important concepts related to our work are described: agents and trust. In Section 4 the trust model is presented. Later in section 5 the multi-agent architecture proposed to manage trustworthy KMS is described. In Section 6 a prototype developed to evaluate our architecture is explained in order to illustrate how it could be used. Section 7 describes a preliminary experiment carry out to test this prototype. Section 8 outlined related work and finally, conclusions are presented in Section 9.

COMMUNITIES OF PRACTICE

Intellectual capital and knowledge management are currently growing since knowledge is a critical factor for an organization's competitive advantage (Kautz, 2004). This growth determines organizations' performance by studying how well they manage their most critical knowledge. However, to manage this critical knowledge it has to be known what knowledge is, and although there is no consensus about a knowledge concept (Kakabadse, et al, 2001), there are several definitions of knowledge as in (Ackoff, 1989) and (Davenport et al, 1998). In our case, knowledge is going to be understood as in (Ackoff, 1989), that is, as an appropriate collection of information, such that its intent is to be useful. In order to manage knowledge an important instrument are communities (Gebert et al., 2004; Malhotra, 2000). A community can be defined as a group of socially interacting persons who are mutually tied to one another and regularly meet at a common place (Hillery, 1955). The development of Internet and groupware technologies led to a new kind of community "virtual communities" where members can or not meet one another face to face and they may exchange words and ideas through the mediation of computers networks (Geib et al., 2004).

This type of communities can be divided regarding 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 to support their professional tasks. An special case of professionally-oriented communities are the "Communities of Practice" (CoPs) defined by Wenger et al. (2002) as groups of people who share a concern, a set of problems, or a passion about a topic, and who deepen their knowledge

and expertise in this area by interacting on an ongoing basis.

Regarding a knowledge point of view CoPs share values, beliefs, languages, and ways of doing things many companies report that CoPs help reduce problems due to lack of communication, and save time by "working smarter" (Wenger, 2002). Millen et al. (2002) discuss the costs and benefits of CoPs in a study of seven large, geographically dispersed organizations. The study indicates benefits on individual, community and organizational level, like increased access to experts and information resources, increased idea generation and better problem solving, to indications of more successfully executed projects and product innovations. Moreover, individuals are frequently more likely to use knowledge built by their community team members than that created by members outside their group (Desouza et al, 2006), that is, they are more likely to share knowledge with people they trust. However, since in current CoPs people are usually geographically dispersed they do not have a face to face communication and this situation could be a problem since the level of trust between members can decrease and in consequence there could be a lower level of sharing knowledge.

On the other hand, even though CoPs are a focus of knowledge sharing hardly ever there is any quality control of the knowledge generated in the community. Our proposal is to use agent's technology to foster knowledge exchange at communities of practice and to evaluate the suitability of knowledge shared.

AGENTS AND TRUST

Because of the importance of knowledge management, tools to support some of the tasks related to knowledge management have been developed. Different techniques are used to implement these tools. One of them, which is providing to be quite useful, is that of intelligent agents (van-Elst et al, 2003). Software agent technology can monitor and coordinate events, meetings and disseminate information (Balasubramanian et al, 2001). Furthermore, agents are proactive; this means they act automatically when it is necessary. The autonomous behavior of the agents is critical to the goal of this research since agents help to reduce the amount of work that employees have to perform, for instance searching information in a knowledge base. On the other hand one of the main advantages of the agent paradigm is that 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 (Wooldridge & Ciancarini, 2001). This foundation has led to an increasing interest in social aspects such as motivation, leadership, culture or trust (Fuentes et al, 2004). Our research is related to this last concept of "trust" since artificial agents can be made more robust, resilient and effective by providing them with trust reasoning capabilities.

For agents to function effectively in a community, they must ensure that their interactions with the other agents are trustworthy. For this reason it is important that each agent is able to identify trustworthy partners with which they should interact and untrustworthy correspondents with which they should avoid interaction. The stability of a community depends on the right balance of trust and distrust.

In literature we found several trust and reputation mechanisms that have been proposed to be used in different domains such as e-commerce (Zacharia et al, 1999), peer-to-peer computing (Wang & Vassileva, 2003), recommender systems (Schafer et al, 1999), etc. In next section we describe the trust model that we propose to be used in our multi-agent architecture.

TRUST MODEL

One of our aims is to provide a trust model based on real world social properties of trust in Communities of Practice (CoPs). An interesting fact is that members of a community are frequently more likely to use knowledge built by their community team members than those created by members outside their group (Desouza et al, 2006). This factor occurs because people trust more in the information offered by a member of their community than in that supplied by a person who does not belong to that community. Of course, the fact of belonging to the same community of practice already implies that these people have similar interests and perhaps the same level of knowledge about a topic. Consequently, the level of trust within a community is often higher than that which exists outside the community. As a result of this, as is claimed in (Desouza et al, 2006), knowledge reuse tends to be restricted within groups. Therefore, people, in real life in general and in companies in particular, prefer to exchange knowledge with "trustworthy people" by which we mean people they trust. For these reasons we consider the implementation of a mechanism in charge of measuring and controlling the confidence level in a community in which the members share information to be of great importance.

Most previous trust models calculate trust by using the users' previous experience with other users but when there is no previous experience, for instance, when a new user arrives to a community, these models cannot calculate a reliable trust value. We propose calculating trust by using four factors that can be stressed depending on the circumstances. These factors are:

- *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 (Wasserman & Glaskiewics, 1994). 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 (Dillenbourg, 1999). In an enterprise this position can be established in different ways by, for instance, using an organizational diagram or classifying the employees according to the knowledge that a person has, as can be seen in Allen's proposal in (Allen, 1984), which distinguishes between:
  - Technological gatekeepers, defined as those actors who have a high level of knowledge interconnectedness with other local firms and also with extra-community sources of knowledge. These basically act by channeling new knowledge into the community and diffusing it locally.
  - External stars, which are highly interconnected with external sources of knowledge but have hardly any interaction with other local firms.

  Such different positions inevitably influence the way in which knowledge is acquired, diffused and eventually transformed within the local area. Because of this, as will later be explained, this factor will be calculated in our research by taking into account a weight that can strengthen this factor to a greater or to a lesser degree.
- *Expertise*: This term can be briefly defined as the skill or knowledge that a person who knows a great deal about a specific thing has. This is an important factor since people often trust experts more than novice employees. In addition, "individual" level knowledge is embedded in the skills and competencies of the researchers, experts, and professionals working in the organization (Nonaka & Takeuchi, 1995). The level of expertise that a person has in a company or in a CoP could be calculated from his/her CV or by considering the amount of time that a person has been working on a topic. This is data that most companies are presumed to have.
- *Previous experience*: This is a critical factor in rating a trust value since, as was mentioned in the definitions of trust and reputation, previous experience is the key value through which to obtain a precise trust value. However, when previous experience is scarce or it does not exist humans use other factors to decide whether or not to trust in a person or a knowledge source. One of these factors is intuition.
- *Intuition*: This is a subjective factor which, according to our study of the state of the art, has not been considered in previous trust models. However, this concept is very important because when people do not have any previous experience they often use their "intuition" to decide whether or not they are going to trust something. Other authors have called this issue "indirect reputation or prior-derived reputation" (Mui et al, 2002). 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 (Mui et al, 2002). We have tried to model intuition according to the similarity between personal profiles: the greater the similarity between one person and another, the greater the level of trust in this person as a result of intuition.

By taking all these factors into account, we have defined our own model with which to rate trust in CoPs, and this is summarized in Figure 1.

The main goal of this model is to rate the level of confidence in an information source or in a provider of knowledge in a CoP.

As the model will be used in virtual communities where people are usually distributed in different locations we have implemented a multi-agent architecture in which each software agent acts on behalf of a person and each agent uses this trust model to analyze which person or piece of knowledge is more trustworthy.

As the number of interactions that an agent will have with other agents in the community will be low in comparison with other scenarios such as auctions we cannot use trust models which need a lot of interactions to obtain a reliable trust value; it is more important to obtain a reliable initial trust value and it is for this reason that we use position, expertise and intuition.

As observed in Figure 1, we use four factors to obtain a trust value, but how do we use these factors? We have classified these four factors into two groups: objective factors (position and expertise) and subjective factors (intuition and previous experience). The former are given by the company or community and the latter depend on the agent itself and the agent's experience in time. There are four different ways of using these factors, which depend upon the agent's situation (see Figure 2).

- If the agent has no previous experience, for instance because it is a new user in the community, then the agent will use its intuition and the position and expertise of other agents to discover which other agents it can trust.
- When the agent has previous experience obtained through interactions with other agents but this previous experience is low (low number of interactions), the agent calculates the trust value by considering the intuition value and the previous experience value. For instance, if an agent A has a high experience value for agent B because it interacted with B successfully several times but agent A has a low intuition value for agent B (profiles are not very similar), then

Figure 1. Trust Model

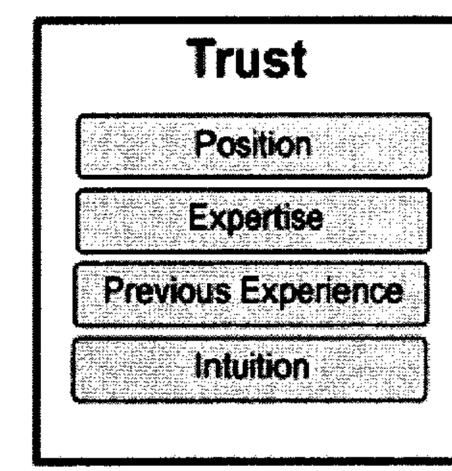
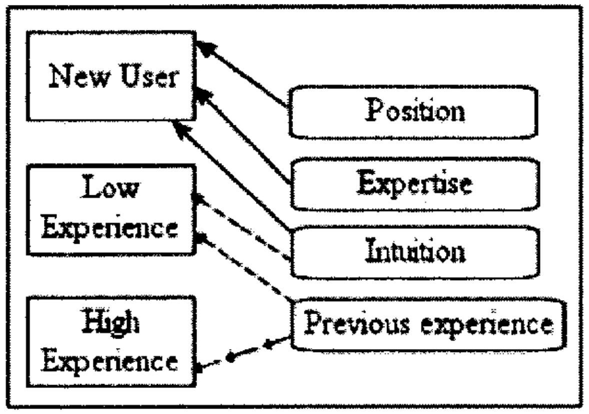


Figure 2. Using the trust model



agent A reduces the value obtained through experience. In this case the agent does not use position and expertise factors (objective factors) because the agent has its own experience and this experience is adjusted with its intuition which is subjective and more personalized.
- When the agent has enough previous experience to consider that the trust value obtained is reliable, then the agent only considers this value.

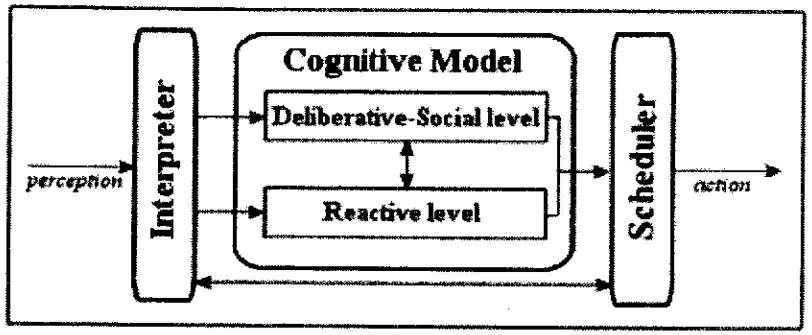
MULTI-AGENT ARCHITECTURE

In order to give support to CoPs, we have designed a multi-agent architecture that uses the trust model explained in previous section with the goal of recommending trustworthy knowledge in CoPs and therefore fostering the reuse of information generated in these communities. Therefore, we can say that the goals of this architecture are:

- Assists members in identifying trustworthy entities.
- Gives artificial agents the ability to reason about the trustworthiness of other agents or about a knowledge source.
- Encourages knowledge exchange between the community members.
- Provides the confidence necessary to foster the usage of information and knowledge of the community.

Taking these facts into account, we propose a multi-agent architecture which is composed of two levels (see Figure 3): a reactive level and a Deliberative-Social level. The reactive level is considered by other authors as a typical level that a multi-agent system must have (Ushida et al, 1998)(Ushida, 1998). 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 (multi-agent systems) are composed of several individuals, interactions between them and plans constructed by them. The social level is only considered in those systems that try to simulate social behaviour or those that represent a more generic architecture which has been prepared to represent this or other behaviour. Since we wish to emulate human feelings such as trust, reputation and even intuition we have added a social part that considers the social aspects of a community which takes into account the opinions and behaviour of each of the members of that community. Other previous works have also added a social level. For instance, in (Imbert & de Antonio, 2005)(Imbert, 2005) the authors emulate human emotions such as fear, thirst or bravery but they use an architecture which is

Figure 3. General architecture



made up of three levels: reactive, deliberative and social. In our case the deliberative and the social level are not separate levels because we realised that plans created in the deliberative part involve social interactions so we considered that in our case it would be more efficient to define a level composed of two parts (Deliberative-Social level) instead of considering two separated levels.

- **Reactive level:** This is the agent's capacity to perceive changes in its environment and to respond to these changes at the precise moment at which they happen. It is in this level when an agent will execute the request of another agent without any type of reasoning.
- **Deliberative-Social level:** The agent has a type of behaviour which is orientated towards objectives, that is, it takes the initiative in order to plan its performance with the purpose of attaining its goals. In 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 fulfil its objectives. In this level we have individual goals which refer to the deliberative part and social goals or cooperative goals which refer to the social part.

Two further important components of our architecture are the *Interpreter* and the *Scheduler*. The former is used to perceive the changes that take place and to decide which level must take the initiative depending on the event that the agent perceives. The scheduler indicates how the actions should be scheduled and executed.

Each of the levels of our architecture is described in the following subsections.

Reactive Level

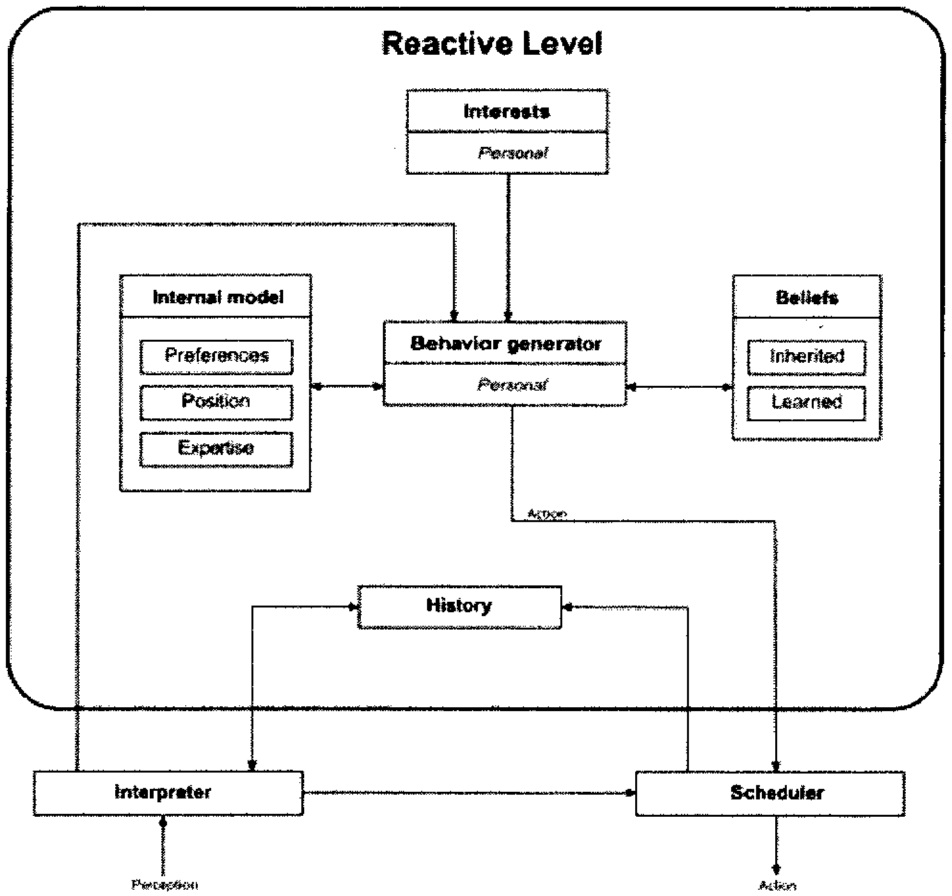
This level must respond at the precise moment at which an event has been perceived (see Figure 4).

For instance when an agent is consulted about its position within the organization or when a user wishes to send the system simple answers. This level is formed of the following modules:

Internal model: As an agent represents a person in a CoP this model stores the user's features, these features will be consulted by other agents in order to calculate trust values. Therefore, this module stores the following parts in the user profile:

- *Expertise.* This term has been explained in the Trust Model in section 4.
- *Preferences.* In this part we try to represent user preferences by using, for example, the Felder-Silverman test which tell us whether the agent is representing a visual user (one who prefers

Figure 4. Reactive level



visual representations of presented information-pictures, diagrams, flow charts,...), a verbal user (who prefers written and spoken explanations) or another kind of user that the Felder-Silverman model supports (Felder & Silverman, 1988; Felder, 1996).

- *Position.* Explained in the Trust Model section.

Behaviour generator: This component is necessary for the development of this architecture since it has to select the agent's behaviour. This behaviour is defined on the basis of the agent's beliefs.

Interests: These are individual interests which represent the user's needs.

History: This component stores the interactions of the agents with the environment.

Beliefs: The beliefs module is composed of inherited beliefs and lessons learned from the agent itself. Inherited beliefs are the organization's beliefs that the agent receives. Examples of this might be an organizational diagram of the enterprise or the philosophy of the company or community. Lessons learned are the lessons that the agent obtains while it interacts with the environment. This interaction can be used to establish parameters in order to know what the agent can trust (agents or knowledge sources).

Deliberative-Social Level

In this level the agent's behaviour is based on goals, that is, the agent has several defined goals and it tries to achieve these goals by scheduling plans. Due to the fact that we are trying to represent human behaviour in CoPs, it is necessary to bear in mind that this human behaviour must benefit the whole community. Therefore, the agent has to deliberate about its individual goals but it must also act by taking community goals and the community's profit into account. That is why we have considered a social and a deliberative part. The former tries to achieve social goals (community goals) and the latter is more focused upon achieving individual goals.

In this level the agent obtains information about the environment and, by taking into account its interests and intuitions, it decides which plan is the best to achieve its goals (see Figure 5).

The components of the Deliberative-Social architecture are:

Interests: This component represents community interests. These interests are created when the community comes into being. There are some interests that all communities may share such as:

- Maintaining a constant collaboration of community members.
- Identifying and maintaining experts in the community
- Keeping community knowledge updated
- Maintaining a trustworthy environment in which community members share trustworthy knowledge.

There are also Personal Interests which influence the whole community such as sharing suitable knowledge.

Beliefs: This module represents a view that the agent has of the environment. In our case these beliefs are composed of the idea that the agent has of the communities and their members. For instance, in this module there is information about the community's topics, in which areas other members are working, etc.

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

Plans Generator: This module is in charge of evaluating how a goal can be attained and which plans are most convenient if this goal is to be achieved. We should recall that plans are a specification of the actions that an agent may carry out in order to attain its goals.

Intuitions: Intuitions are beliefs that have not been verified but which an agent thinks may be true. According to Mui et al. (2002)(Mui, 2002) intuition has not yet been modelled by agent systems. In this work we have tried to adapt this concept by comparing the agents' profiles (as we mentioned in Section 4) to obtain an initial value of intuition that can be used to form a belief about an agent when the intuition is proved be true. This is another important feature taken into account to calculate a trust value, since when an agent has little o null interaction with another; the agent will use this value to have a value of trust as it was previously explained.

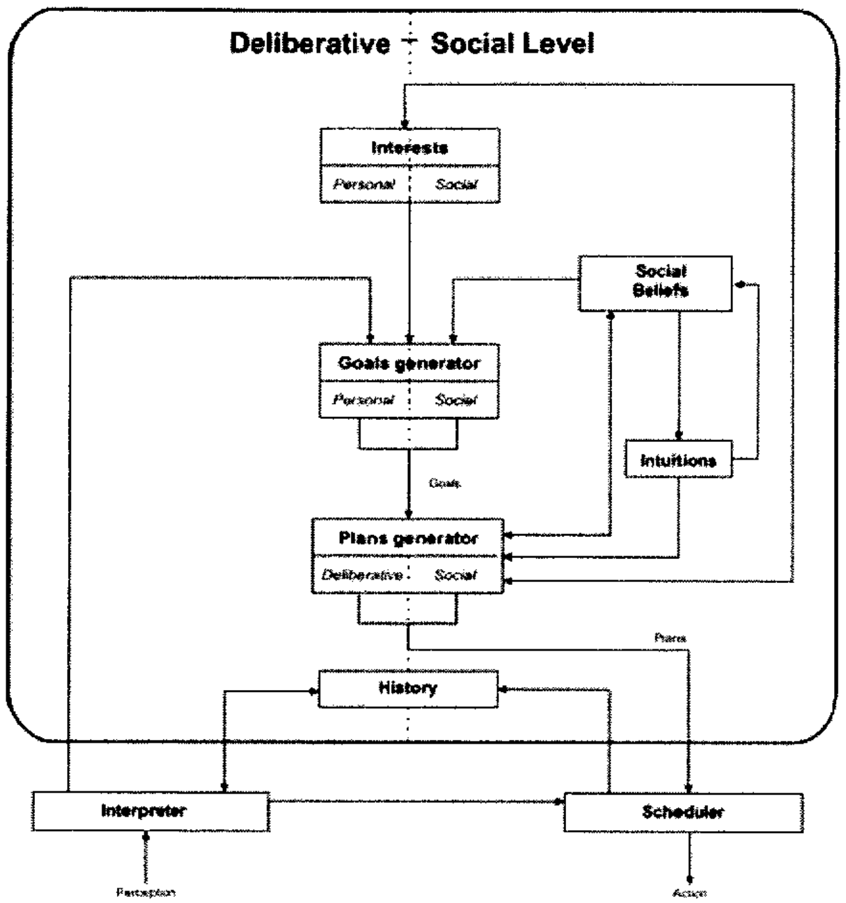
History: This component stores the interactions of the agents with the environment.

In the following section, we will describe a prototype developed to validate our architecture.

PROTOTYPE

In order to test our architecture we have developed a prototype system into which people can introduce documents and where these documents can also be consulted by other people. The goal of this prototype is to allow software agents to help employees to discover the information that may be useful to them thus decreasing the overload of information that employees often have and strengthening the use of knowledge bases in enterprises. In addition, we try to avoid

Figure 5. Deliberative-social level



the situation of employees storing valueless information in the knowledge base.

One feature of this system is that when a person searches for knowledge in a community, and after having used the knowledge obtained, that person then has to evaluate the knowledge in order to indicate whether:

- The knowledge was useful.
- How it was related to the topic of the search (for instance a lot, not too much, not at all).

To design this prototype we have designed a *User Agent* and a *Manager Agent*. The former is used to represent each person that may consult or introduce knowledge in a knowledge base. Therefore, the *User Agent* can assume three types of behavior or roles similar to the tasks that a person may carry out in a knowledge base. 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 **Provider**.
- The person uses knowledge previously stored in the community. Then, the User Agent will be considered as a **Consumer**.

- The person helps other users to achieve their goals, for instance by giving an evaluation of certain knowledge. In this case the role is of a **Partner**. So, Figure 6 shows that in community 1 there are two User Agents playing the role of Partner, one User Agent playing the role of Consumer and another being a Provider.

The second type of agent within a community is called the *Manager Agent* (represented in black in Figure 6) which must manage and control its community.

The prototype provides the options of using community documents and when the documents are used, reputation values can be modified. An user can also propose new topics in the community, etc.

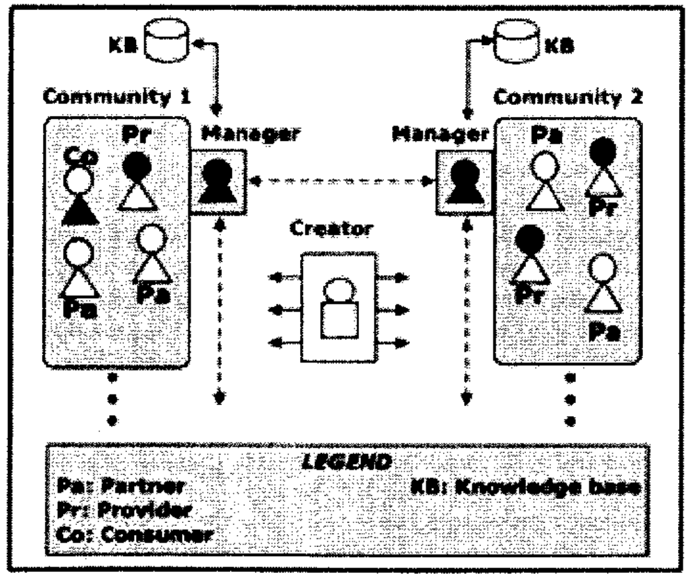
In order to make it easier to search for documents in a community, users can choose one topic from those which are available in the community and the user agent will try to find documents about this topic.

The general idea is to consider those documents which came from trustworthy knowledge sources according to the user's opinion or needs. In order to discover which knowledge sources are trustworthy the user agents will use the trust model. Depending on the context, this trust model can be used in different ways. We are going to consider how the trust model is applied in different situations. First, when agents have previous experience this means that user agents have previously interacted with a knowledge source and they have some feedback (trust values in our case) about it. The second scenario is a more complicated situation in which the agents have no previous experience and therefore do not have trust values for other user agents.

The way in which we apply these factors in the different contexts is as follows:

1. If the agent has no previous experience, for instance, because it is representing a new user in the community, and its user wants to search for documents relating to a topic T, the user agent follows these steps:
   1.1. The user agent makes a request to the other members of the community in order to discover which user agents have documents about topic T.

Figure 6. Communities of agents

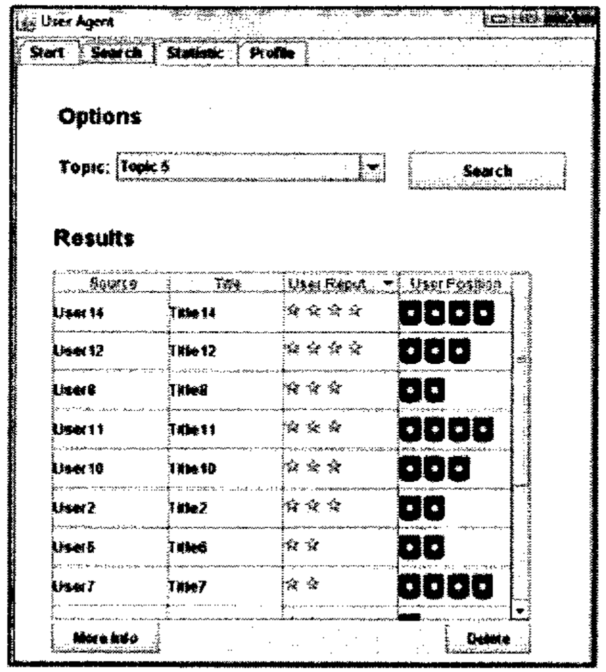


1.2. The user agent stores the id (identification) in a list of those agents which have documents about T.
1.3. For each agent grouping in the list, the user agent calculates a trust value by using the position, expertise and intuition factors. For instance, the user agent might obtain a list with 10 agents that match the request and for each of these agents, the user agent will obtain information about their positions (to discover, for instance, if the agent represents a boss or a newcomer), their levels of expertise in the community area, in our case there are five possible levels (from novice to expert), and their intuition values in relation to the agent that has made the request (with five values from "totally different" to "totally equals"). In this case the intuition level is calculated by comparing user profiles, that is, if the user agent compares two profiles with very similar characteristics, this means that users, represented by user agents, work in the same area, have similar expertise level,..., etc and consequently the trust value will increase because the user agent "senses" that working with this user will be a successful interaction. So, the user agent's list might contain an agent that represents a newcomer user with a high level of expertise and with similar preferences, or a boss with different preferences and a medium level of expertise in the area concerned. Once the agent has obtained all these values, it calculates a general trust value per agent by combining the different factors, obtaining the lowest value when the agent, for instance, represents a rookie newcomer with a profile which is totally different to that of the requester, and obtaining the highest value when the agent represents a boss with a high level of experience and who has a very similar profile to the agent which is making the request.
1.4. The user agent shows the results which are sorted by trust values, that is, the first documents on the list come from the most trustworthy knowledge sources (in this case the most trustworthy agent with the highest trust values). There are other possibilities, depending on user preferences. The user can choose to sort the list by using level of experience, position or level of intuition. At each request the user will receive a list and from each list the user will obtain information about each factor by the use of star icons and shield icons. For instance, as we can see in Figure 7, the results of the request (sorted by reputation) show a large amount of results, and the first one on the list has five stars in the reputation level and four shields in the position level.

2. If there is a small amount of previous experience and this previous experience is not sufficient for the agent to discover whether the other user agent is trustworthy or not then we combine previous experience with the other three factors. So in this context the user agent follows the following steps when looking for documents about a topic T:
2.1. The user agent makes a request to the members of the community in order to discover which user agents have documents about topic T.
2.2. For each agent (in the requested group) that our agent has previously interacted with, it uses the four factors (position, intuition, expertise and previous experience) to calculate a trust value by using (1).

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

Figure 7. Showing and sorting results



where 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. In this case the domain of the community which the agent wishes to join.

P_j is the value assigned to a person's position. This position is defined in the agent's internal model of the reactive architecture described in Section 4.1.

I_{ij} denotes the intuition value that agent i has in agent j which is calculated by comparing each user's profile.

In addition, previous experience should also be calculated. When an agent i consults information from another agent j, the agent i should evaluate how useful this information was. This value is called QC_{ij} (Quality of j's Contribution in the opinion of i). To attain the average value of an agent's contribution, 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.

2.3. For each agent in the group (the results group) that the agent has no previous experience it calculate a trust value as we mentioned in 1.3.
2.4. The user agent shows the results, which are sorted by trust or quality values as in the previous situation.

3. If the user agent has enough previous experience (this is considered when an agent has interacted many times with another. This number of interactions depends on

a threshold that can be adjusted to each domain) then the user agent calculates the trust value by only using the previous experience factor. In this case we only consider this factor (experience) because this is the principal factor that humans usually consider when they have to trust somebody/something. That's why this concept is the base of all trust models described in literature as it will be explained in section 7. In this context the user agent follows the following steps when looking for documents about a topic T:

3.1. The user agent follows step 2.1
3.2. For each agent in the group (the results group) the user agent calculates a trust value by using the previous experience factor that is, by using (2) which is the last part of formula (1),

$$(\sum_{j=1}^{n} QC_{ij})/n \qquad (2)$$

3.3. The user agent follows step 2.3.
3.4. The user agent shows the results which are sorted by trust or quality values.

These are three possible scenarios that illustrate how the trust model is used. When a person inserts a document in the community, s/he inserts the document and a quality value for that document. If another person uses that document, after using that document, the person who requested it must evaluate its quality. The User Agent compares the value given by the owner with the value given by the consumer to discover whether the two users have the same opinion about the document. If this is so then the previous experience value for the other user increases and if the opposite is true then the previous experience value is decreased. That is, if a user A thinks that a document D has a quality value of 8 and another user B, after using D, thinks that the document has a quality value of 2, the trust value that user B has for user A is decreased. This manner of rating trust helps to detect a problem which is increasing in companies or communities in which employees introduce not valuable information because they are rewarded if they contribute with knowledge in the community. Thus, if a person introduces documents that are not related to the community with the aim of obtaining rewards, the situation can be detected, because when the other person evaluate those documents or information, the rate of them will be low and the value of previous experience of this person became very low. Therefore, the community agent can detect that there is a "fraudulent" member in the community.

In a previous version of the prototype, when a person introduced a document in the system, s/he did not indicate the quality value of it. In this case the previous experience was calculated only in base to the rate that agent gave. We have introduced this change because we have a reference value for each document that can be used to compare quality values that different users have given to the same document and detect if there are "fraudulent members" and also can be used to sort documents when we have not enough information, at least we will have a quality value given by the person who introduced the document in the community.

The three situations must be applied to each user agent depending on the situation. If a user agent makes a request to search for documents it will receive answers from different user agents and, depending upon the situation between the requester and the other agents, the requester must apply one of the three situation steps and not only one situation for all the agents that have answered. This is shown in Figure 8 where agent *A* makes a request to search for documents about topic T and agents *X*, *Y*, *Z* answer because they have documents about T. In this case agent *A* applies situation 1 to agent *X* because agent *X* is not a known agent, and situation 3 to agents *Y* and *Z* because it has already interacted with both agents on previous occasions.

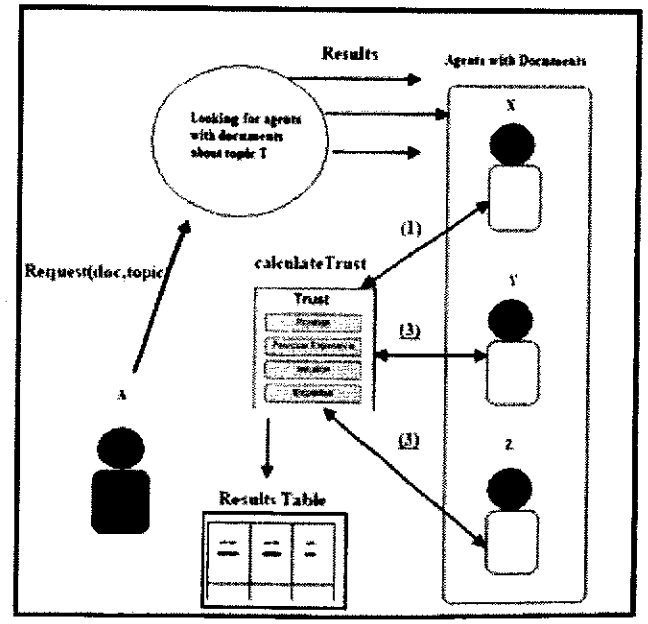
EVALUATION OF THE PROTOTYPE

Once the prototype has been finished we have evaluated it. To do this, different approaches can be followed, from a multi-agent point of view or from a social one. First of all we have focused on the former and we are testing the most suitable number of agents advisable for a community. Therefore, several simulations have been performed. As result of them we found that:

- The maximum number of agents supported by the Community Manager Agent when it receives User Agents' evaluations is approximately 800. When we tried to work with 1000 agents for instance, the messages were not managed conveniently. However, we could see that the Manager Agent could support a high number of petitions, at least, using simpler behavior.
- On the other hand, if we have around 10 User Agents launched, they need about 20 o more interactions to know all agents of the community. If a User Agent has between 10 and 20 interactions with other members it is likely that it interacts with 90% of members of its community, which means that the agent is going to know almost all the members of the community. Therefore, after several trials we detected that the most suitable number of agents for one community was around 10 agents and they needed a average of 20 interactions to know (to have a contact with) all the members of the community, which is quite convenient in order to obtain its own value of reputation about other agent.

All these results are being used to detect whether the exchange of messages between the agents is suitable, and to see if the information that we propose to be taken into account to obtain a trustworthy value of the reputation of each agent is enough, or if more parameters should be considered. Once this validation is finished we need to carry out further research to answer one important and tricky question, which is how the usage of this prototype affects the performance of a community.

Figure 8. Mechanism through which to obtain trustworthy documents by using the model



Copyright © 2009, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

RELATED WORK

This research can be compared with other proposals that use agents and trust in knowledge exchange. With regard to trust, in models such as eBay (1995)(ebay, 1995) and Amazon (1996)(Amazon, 1996), which were proposed to resolve specific situations in online commerce, the ratings are stored centrally and the reputation value is computed as the sum of those ratings over six months. Thus, reputation in these models is a single global value. However, these models are too simple (in terms of their trust values and the way in which they are aggregated) to be applied in open multi-agent systems. For instance, in (Zacharia et al, 1999)(Zacharia, 1999) the authors present the Sporas model, a reputation mechanism for loosely connected online communities where, 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 Histos which is a more personalized system than Sporas and is orientated towards highly connected online communities. In (Sabater & Sierra, 2002)(Sabater, 2002) 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. Carbó et al (2003)(Carbó, 2003) presents the AFRAS model, which is based on Sporas but uses fuzzy logic. 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 (Abdul-Rahman & Hailes, 2000)(Abdul-Rahman, 2000) the authors propose a model which allows agents to decide which agents' opinions they trust more and to propose a protocol based on recommendations. This model is based on a reputation or word-of-mouth mechanism. The main problem with this approach is that every agent must maintain rather complex data structures which represent a kind of global knowledge about the whole network.

Barber and Kim (2004) present a multi-agent belief revision algorithm based on belief networks (Barber, 2004). 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. Some of these trust and reputation models are summarized in Table 1.

In (Huynh et al, 2004)(Huynh, 2004) the authors present a trust and reputation model which integrates a number of information sources in order to produce a comprehensive assessment of an agent's likely performance. In this case the model uses four parameters to calculate trust values: interaction trust, role-based trust, witness reputation and certified reputation. We use certified reputation when an agent wishes to join a new community and uses a trust value obtained in other communities, but in our case this certified reputation is made up of the four previously explained factors and is not only a single factor.

Also, works such as (Guizzardi et al, 2004)(Guizzardi, 2004) use the term 'Community' to support knowledge management but it is not used a specific trust model for communities.

The main differences between these reputation models (summarized in Table1) and our approach are that these models need an initial number of interactions to obtain a good reputation value and it is not possible to use them to discover whether or not a new user can be trusted. A further difference is that our approach is orientated towards collaboration between users in CoPs. Other approaches are more orientated towards competition, and most of them are tested in auctions.

Table 1. Other trust and reputation models

Model	Authors	Reputation Management	Features
ebay	-	Global values	Simple values obtained through interactions
Sporas	Zacharia	Global values	Reduces changes when reputation is very high Most recent reputation values are the most important
Histos	Zacharia	Pair wise ratings in the system as a directed graph	Divides Reputation into three dimensions: Individual, Social and Ontological
Regret	Sabater and Sierra	Decentralized values	Most recent reputation values are the most important Presents a witness reputation component
Afras	Carbó and Molina	Decentralized values	Based on BDI agents Based on Sporas model but using fuzzy logic Compares and combines fuzzy sets
Fire	T. Dong Huynh and Nicholas R. Jennings	Decentralized values	Four main components: interaction trust, role-based trust, witness reputation, and certified reputation

CONCLUSION

Communities of practice have the potential to improve organizational performance and facilitate community work. Because of this we consider it important to model people's behavior within communities with the purpose of imitating the exchange of information that are produced in those communities. Therefore, we are attempting to encourage the sharing of information in organizations by using CoPs and knowledge bases. To do this we have designed a multi-agent three-layer architecture where the artificial agents use similar parameters to those of humans in order to evaluate knowledge and knowledge sources. These factors are: reputation, expertise, position, previous experience and even intuitions.

This approach implies several advantages for organizations as it permits them to identify the expertise of their employees and to measure the quality of their contributions. Therefore, it is expected a greater exchange and reuse of knowledge.

In addition, this work has illustrated how the architecture can be used to implement a prototype. The main functionalities of the prototype are:

- Controlling those employees who try to introduce valueless knowledge with the goal of obtaining some profit such as points, incentives, rewards, etc.
- Providing the most suitable knowledge for the employee's queries according to the employee features and needs.
- Detecting the expertise of the employees within an organization.

All these advantages provide organizations with a better control of their knowledge repositories which will have more trustworthy knowledge and it is consequently expected that employees will feel more willing to use it.

ACKNOWLEDGMENT

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

REFERENCES

Abdul-Rahman, A. and Hailes, S., (2000), "Supporting Trust in Virtual Communities". Proceedings of the 33rd Hawaii International Conference on Systems Sciences (HICSS'00), Vol. 6.

Ackoff, R., (1989), "From Data to Wisdom". Journal of Applies Systems Analysis. Vol. 16, pp. 3-9.

Allen, T., (1984), "Managing the Flow of Technology: Technology Transfer and the Dissemination of Technological Information within the R&D Organization", Cambridge, MA: MIT Press.

Amazon (1996). "URL: http://www.amazon.com".

Balasubramanian, S., Brennan, R., Norrie, D., (2001), "An Architecture for Metamorphic Control of Holonic Manufacturing Systems". Computers in Industry, Vol. 46(1), pp. 13-31.

Barber, K. and Kim, J., (2004), "Belief Revision Process Based on Trust: Simulation Experiments". 4th Workshop on Deception, Fraud and Trust in Agent Societies, Montreal Canada, pp. 1-12.

Carbó, J., Molina, M., Dávila, J., (2003), "Trust Management through Fuzzy Reputation". International Journal of Cooperative Information Systems. Vol. 12(1), pp. 135-155.

Davenport, P., (1998), "Working Knowledge: How Organizations Manage What They Know". Boston, MA, Project Management Institute, Harvard Business School Press.

Desouza, K., Awazu, Y., Baloh, P., (2006), "Managing Knowledge in Global Software Development Efforts: Issues and Practices". IEEE Software, pp. 30-37.

Dillenbourg, P., (1999), "Introduction: What Do You Mean By 'Collaborative Learning'?." Collaborative Learning Cognitive and Computational Approaches. Dillenbourg (Ed.). Elsevier Science.

eBay (1995). "URL: http://www.ebay.com".

Felder, R. and Silverman L, (1988), "Learning and Teaching Styles in Engineering Education". Engineering Education. Vol. 78(7), pp. 674-681.

Felder, R. M. (1996). "Matters of Style". ASEE Prism. Vol. 6(4), pp. 18-23.

Fuentes, R., Gómez-Sanz, J., Pavón, J. (2004). "A Social Framework for Multi-agent Systems Validation and Verification". Wang, S. et al Eds. ER Workshops, Springer Verlag, LNCS 3289, pp. 458-469.

Gebert, H., Geib, M., Kolbe, L., Brenner, W., (2003), "Knowledge-enabled Customer Relationship Management - Integrating Customer Relationship Management and Knowledge Management Concepts". Journal of Knowledge Management. Vol. 7(5), pp. 107-123.

Geib, M., Braun, C., Kolbe, L., Brenner, W., (2004). Measuring the Utilization of Collaboration Technology for Knowledge Development and Exchange in Virtual Communities. 37th Hawaii International Conference on System Sciences 2004 (HICSS-37), Big Island, Hawaii, IEEE Computer Society, Vol. 1, pp. 1-10.

Guizzardi, R., Perini, A., Dignum, V., (2004), «Providing Knowledge Management Support to Communities of Practice through Agent-Oriented Analysis». Proceedings of the 4th International Conference on Knowledge Management (I-KNOW), Granz, Austria.

Hillery, G., (1955), «Definitions of Community: Areas of Agreement», Rural Sociology, Vol. 20, pp. 118-125.

Huynh, T., Jennings, N., Shadbolt, N., (2004), «FIRE: An Integrated Trust and Reputation Model for Open Multi-agent Systems». Proceedings of the 16th European Conference on Artificial Intelligence (ECAI).

Imbert, R., and de Antonio, A., (2005), «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, pp.152-165.

Kakabadse, N., Kouzmin, A., Kakabadse, A., (2001), «From Tacit Knowledge to Knowledge Management: Leveraging Invisible Assets». Journal of Knowledge and Process Management, Vol. 8(3), pp. 137-154.

Kautz, H. (2004), «Knowledge Mapping: A Technique for Identifying Knowledge Flows in Software Organizations», EuroSPI, pp. 126-137.

Malhotra, Y. (2000), «Knowledge Management and Virtual Organizations», IDEA Group publishing, Hershey.

Millen, D., Fontaine, M., Muller, M., (2002), "Understanding the benefits and costs of communities of practice". Communications of the ACM., Vol. 45(4), pp. 69-73.

Mui, L., Halberstadt, A., Mohtashemi, M., (2002), "Notions of Reputation in Multi-Agents Systems: A Review". International Conference on Autonomous Agents and Multi-Agents Systems (AAMAS), pp. 280-287.

Nonaka, I. and Takeuchi, H., (1995), "The Knowledge Creation Company: How Japanese Companies Create the Dynamics of Innovation", Oxford University Press.

Sabater, J. and Sierra, C., (2002), "Social REGRET, a Reputation Model based on social relations", Proceedings of the Fifth International Conference on Autonomous Agents. Vol. 3(1), pp. 44-56.

Schafer, B. J., Konstan, A., J., Riedl, J. (1999). "Recommender Systems in E-Commerce", 1st ACM Conference on Electronic Conference (EC), pp. 158-166.

Ushida, H., Hirayama, Y., Nakajima, H., (1998), "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), Madison, Wisconsin, USA, pp. 8-37.

van-Elst, L., Dignum, V., Abecker, A., (2003), "Agent-Mediated Knowledge Management". International Simposium AMKM, Stanford, CA, USA, Springer, pp. 1-30.

Wang, Y., Vassileva, J., (2003), "Trust and Reputation Model in Peer-to-Peer Networks". Proceedings of the 3rd International Conference on Peer-to-Peer Computing.

Wasserman, S. and Glaskiewics, J., (1994), "Advances in Social Networks Analysis". Sage Publications.

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

Wenger, E., McDermott, R., Snyder, W., (2002), "Cultivating Communities of Practice", Harvard Business School Press.

Wooldridge, M., Ciancarini, P., (2001), Agent-Oriented Software Engineering: The State of the Art.

Zacharia, G., Moukas, A., Maes, P. (1999). "Collaborative Reputation Mechanisms in Electronic Marketplaces". In 32nd Annual Hawaii International Conference on System Science (HICSS-32).

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

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

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

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