International Journal on Advances in Life Sciences, vol 1 no 2&3, year 2009, http://www.iariajournals.org/life_sciences/

Encouraging the Participation and Knowledge Reuse in Communities of Practice by Using a Multi-Agent Architecture and a Trust Model

Aurora Vizcaíno, Juan Pablo Soto, Javier Portillo-Rodríguez, Mario Piattini
Alarcos Research Group – Institute of Information Technologies & Systems
Dep. of Information Technologies & Systems – Escuela Superior de Informática
University of Castilla – La Mancha
Ciudad Real, Spain
{aurora.vizcaino, javier.portillo, mario.piattini}@uclm.es, jpsotob@gmail.com

Abstract — **This paper proposes a multi-agent architecture based on the concepts of communities of practice and trust to manage knowledge management systems. The main goal of this proposal is to assist community of practice members in deciding what or who to trust and in this way attempt to foster the reuse of information in organizations which use knowledge management systems. One contribution of this work is a trust model which takes into account certain factors that human beings consciously or unconsciously use when they have to decide whether or not to trust in something or somebody. Moreover, in order to illustrate how the model can be used, a prototype with which to recommend documents is also described.**

Keywords — Multi-agent System, Communities of Practice, Trust, Knowledge Management.

I. INTRODUCTION

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 result of this, these systems are sometimes not greatly used 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. For this, 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 Communities of Practice (CoPs). CoPs are becoming increasingly more common in organizations due to the fact they are a means of sharing knowledge [2] [3]. They are frequently defined as groups of people who share a concern, a set of problems, or a passion about a topic and who extend their knowledge and expertise in this area by interacting on an ongoing basis [4]. However, CoPs members are ever-increasingly distributed throughout different geographic locations. This implies a lack of face-to-face communication which affects certain aspects of interpersonal relationships. For instance, if people never experience face-to-face communication and only use groupware tools to communicate, then trust often decreases [5][33]. This lack of trust makes it more difficult for CoPs members to know which of their fellow-members are more trustworthy. This presents a problem, as in CoPs the main knowledge sources are the members themselves. We thus consider that it is highly important to be able to discover how trustworthy a knowledge source (i.e. another member) is. This knowledge will help members to decide whether or not a piece of knowledge is valuable depending on the knowledge source from which it originates. Therefore, in order to support CoPs members in this task, this paper describes a trust model designed solely for CoPs in which various psychological aspects that a person uses, either consciously or unconsciously, to value whether another person is trustworthy have been considered. This model has been used in the implementation of a prototype in which software agents make recommendations to users about what documents are most relevant to them according to their preferences and trust in knowledge sources.

In the following section we describe the multi-agent architecture proposed. Later, the next section describes the trust model that we propose. Section Four explains the details of how this model was implemented in a prototype. Section Five outlines related work. Finally, in Section Six, our conclusions are summarized.

II. A MULTI-AGENT ARCHITECTURE

The multi-agent architecture proposed is composed of two levels (see Figure 2): reactive and deliberative-social. The reactive level is considered by other authors to be a typical level that a Multi-Agent System (MAS) must have [6]. A deliberative level is often also considered as a typical level but a social level is not frequently considered in an explicit way, despite the fact that these systems (MAS) are composed of several individuals, the interactions between them and the plans constructed by them. The social level is only considered in those systems that attempt to simulate social behavior. Since we wish to emulate human feelings such as trust and intuition when working in CoPs, we have added a social level that considers the social aspects of a community and which takes into account the opinions and behavior of each of the members of that community. Other previous works have also added a social level. For example, Imbert & de Antonio [7] 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 involved 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 subsections.

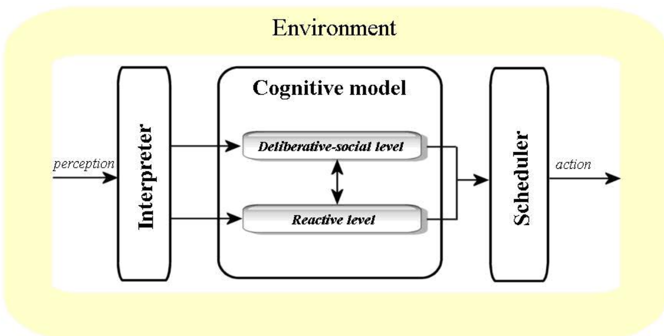

Figure 1. Multi-agent architecture

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

2.1 Reactive level

This is the level in charge of perceiving 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 2):

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 also knowledge sources since they contribute to the CoP with information. Therefore, the model stores the following information, which will be useful in calculating how trustworthy a knowledge source is:

- *Expertise*. This information is an important factor since people often trust experts more than novice employees. The level of expertise that an individual has in a CoP could, for example, be calculated, from his/her CV or by considering the amount of time that a person has been working on a topic.
- *Position*. Employees often consider information that comes from a superior as being more reliable than that which comes from another employee in the same (or a lower) position as him/her [8]. 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 [9]. 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 that the agent is acting on behalf of. Therefore, a person's preferences are stored here.

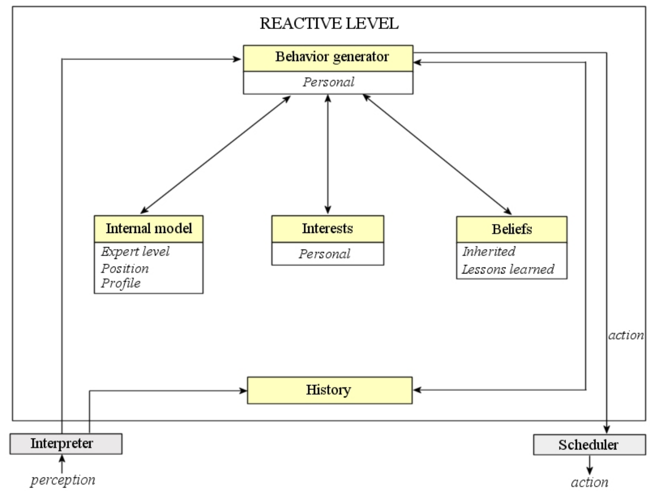

Figure 2. Reactive level

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.

Interests. They are a special kind of beliefs. This component represents individual interests that an agent has about a topic or about a knowledge source.

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

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

2.2 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 3):

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, beliefs about other agents.

Social interests. This is a special type of belief. In this case it is represented interest about other agents.

Intuitions. As we are modelling community members we have attempted to introduce factors into this architecture 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" [10]. 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 [10]. 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* [11], 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.

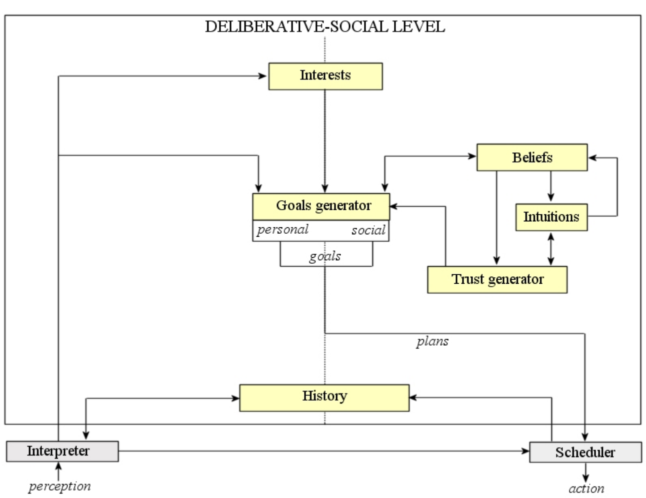

Figure 3. Deliberative-social level

Trust generator. This module is in charge of generating a trust value for the knowledge sources with which an agent interacts in the community. To do this, the trust generator module considers the trust model explained in detail in [12] which considers the information obtained from the internal model and the agent's intuitions.

III. THE TRUST MODEL

It is first important to clarify that this trust model was designed to be used in companies in which CoPs are created as a knowledge management strategy with the goal of sharing knowledge and reusing lessons learnt. The word 'employees' therefore appears in this paper on several occasions, as it is assumed that the final aim of this research is to support companies, enterprises and organizations in general in the creation and use of CoPs as a means of improving their knowledge management.

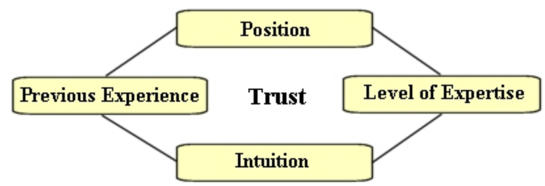

Figure 4. Trust factors

There are many recent proposals for reputation mechanisms and approaches to evaluate trust in P2P systems in general [13, 14], and multi-agent systems in particular [15, 16, 17, 18, 19, 20, 21]. However, there is no universal agreement on the definition of the trust and reputation. Since the main goal of our work is to rate the credibility of information sources and of knowledge in CoPs, it is first necessary to define these two important concepts.

Trust is a complex notion whose study is usually of a narrow scope. This has given rise to an evident lack of coherence among researchers in the definition of trust. For instance Wang & Vassileva define trust as a peer's belief in another peer's capabilities, honesty and reliability based on his/her own direct experiences [13].

Another important concept related to trust is reputation. Several definitions of reputation can be found in literature, such as that of Barber & Kim whom define this concept as the amount of trust that an agent has in an information sources [18], created through interactions with information sources, and that of Mui *et al* [22] which define reputation as a perception a partner creates through past actions about his intentions and norms. This may be considered as a global or personalized quantity [22].

These concepts of trust and reputation are sometimes used interchangeably. However, recent research has shown that there is a clear difference between them, whilst accepting that there is a certain amount of correlation between the two concepts in some cases [23, 24].

In our work we intend to follow the definition given by Wang & Vassileva [13] which considers that the difference between both concepts depends on who has previous experience, so if a person has direct experiences of, for instance, a knowledge source we can say that this person has a trust value in that knowledge.

Many authors consider that trust facilitates problem solving by encouraging information exchange [25]. However, the development of trust in a virtual setting is often more difficult than in co-located meetings [26]. Moreover, the idea of trusting or not trusting in something or somebody is context dependent. For instance, at an auction people may attempt to cheat in order to obtain greater benefits. Furthermore, in a CoP other factors may arise which might be objective and sub-objective. Both types have been considered in this model (see Figure 4), since both are frequently relevant in the personal decision-making processes.

The first is that of the **Position** that a person holds in the organization in which the CoPs exist. This factor will be calculated in our research by considering a weight that can strengthen this factor to a greater or to a lesser degree. This is an objective factor since it is provided or indicated by an exterior entity (for instance, it may be provided by the organization, by the community itself, etc).

Level of Expertise (**LE**): this term can be briefly defined as the skill or knowledge of a person who knows a great deal about a specific thing. This is an important factor since people often trust in experts more than in novice employees. In addition, an "individual" level of knowledge is embedded in the skills and competencies of the researchers, experts, and professionals working in the organization [27].

This factor can be seen as objective or sub-objective according to where this concept originates. For instance if it is specified by the organization it will be considered as objective. However, if its value is provided by the opinion of another agent then it will be seen as a sub-objective value.

Previous experience (**PE**): A trusting decision is based on the truster's relevant prior experiences and knowledge [28, 29]. Experiences and knowledge form the basis of trust in future familiar situations [30]. Consequently, members of CoPs have greater trust in those knowledge sources from which they have previously obtained more "valuable information". Therefore, previous experience increases or decreases trust, and this factor can be very useful in detecting trustworthy knowledge sources in CoPs. In this case this factor is subjective since it depends on a person's opinion.

Intuition (**I**): When people do not have any previous experience they often use their "intuition" to decide whether or not they are going to trust something. In this research, intuition has been modelled according to the similarity between agents' profiles: the greater the similarity between one agent and another, the greater the level of trust. This is, of course, a highly subjective value because it is almost at the same level as a hunch and depends directly on the point of view of each person.

As will later be explained, it is possible to decide to place more importance upon one factor or another according to the setting in which the trust model is used. For this reason, we have pondered each factor with a weight which emphasizes a factor or decreases its importance. An explanation of how to use this model will be shown in the following section.

IV. A PROTOTYPE TO RECOMMEND DOCUMENTS

In order to test the trust model, a prototype with which to recommend documents to CoP members was developed. This prototype allows CoP members to introduce documents relating to different topics. Each time a person uses a document recommended by this tool, that person should evaluate it to enable the prototype to obtain user-feedback.

The prototype was developed by using the software architecture described in section 2, This section will centre on explaining how agents calculate each factor of the trust model explained in the previous section, and which is considered in the following formula:

$$T_{\boldsymbol{ij}} = wp*P_j + we*LE_j + wi*I_{ij} + PE_{ij} \quad (1)$$

Let us then imagine that an agent *i* must evaluate how trustworthy another agent *j* is. It will therefore use Formula (1) in which $T_{\boldsymbol{ij}}$ is the value of *j*'s trust in the eyes of *i*. We shall now describe how each factor of the formula is calculated.

Position

When a new member joins a community that person must indicate his/her position within the organization and his/her software agent will calculate the Position (P) value of that person by using the following formula:

$$P = UPL/NL \quad (2)$$

where UPL is the user's position level and NL is the number of levels in the community.

Therefore, if a community, for instance, has 5 possible position levels then NL=5, and if the new member has a level of UPL=2 then the value of P will be 2/5=0.4. Therefore, the different values of P for a community with five levels will be those shown in Table 1:

TABLE I. EXAMPLE OF POSITION LEVELS

Levels	Values P
1	0.2
2	0.4
3	0.6
4	0.8
5	1

The P values will always be between 0 and 1. Moreover, situations may exist in which P will not been taken into account, for instance in those CoPs in which all the members have the same level or whose members do not wish to consider this criterion. In these cases wp (weight of position) will be zero and position will not be considered in the formula. A further situation exists in which wp is equal to zero. This occurs when the value of the PE > U (U being a threshold which is chosen when creating the community). In this case, the agent will use the following formula to calculate the wp value:

$$wp = int\ (U/PE_{ij})\ \text{being}\ PE_{ij} > 0$$

where U is a threshold of previous experience. PE_{ij} is the value of previous experience of an agent *i* with another agent *j*.

Thus, when PE_{ij} is greater than a particular threshold U, wp will be 0, thus ignoring the position factor. However, when one agent does not have enough PE of another it may use other factors to obtain a trust value. On the other hand, when the agent has had a considerable amount of PE with this agent or with the knowledge that it has provided then it is more appropriate to give more weight to this factor, since PE is the key factor in all trust models, as will be described in Section 4. Therefore, if an agent *j* has a high value of position but most of agent *i*'s previous experience of *j* has not been successful then the position will be ignored. This thus avoids the situation of, for instance, a boss who does not contribute with valuable documents but is considered trustworthy solely because s/he is a boss.

Level of Expertise

As was previously mentioned, this factor is used to represent the level of knowledge and know-how that a person has in a particular domain. In this prototype this factor may change since a person may become more expert in a topic as time goes by.

In this tool, when creating a community the levels of expertise considered is also indicated, for instance: novice, beginner, competent, expert and master. Each time a new member joins a community s/he will indicate the level of expertise that s/he considers him/herself to have. If the members of the community and their level of expertise are known to the creator of the community then that person can introduce them in the tool. Once the level of expertise has been introduced, the user agent will calculate the value for this level by using the following formula:

$$LE_j = L_j/NT + AV_j \qquad (3)$$

where L_j is the level of expertise that was introduced, and NT is the number of levels in the community. The term AV_j is the Adjustment Value for agent *j*. This term is extremely important since it will be used to adjust the experience of each user. This term was introduced with the goal of avoiding two situations:

- That a person either deliberately or mistakenly introduces a level of experience that is not the level that s/he has.
- That, whilst in the community, a person becomes more expert leading to the situation that his/her level of expertise should be adjusted.

Initially AV_j will be 0, and each time a member interacts with a document or information provided by *j* the member will rate this document or information and send this evaluation to an agent called the manager agent which is in charge of managing the community. The manager agent will verify whether the evaluation is negative or positive. If it is positive, then agent *j*'s level of experience can be modified by calculating AV_j as:

$$AV_j = (VL_n - VL_{n-1})/PT \quad (n \neq 1)$$

If it is negative, then:

$$AV_j = -\ (VL_n - VL_{n-1})/PT \quad (n \neq 1)$$

where VL_n is the value that a particular level of experience has. PT is the Promotion Threshold which is used to determine the number of positive rates necessary to promote a superior level of experience. Let us illustrate this with an example. In a community there are four levels with the following values.

TABLE II. POSITION VALUES

Labels	Level(n)	Value(VL)
Novice	1	0
Beginner	2	0.25
Competent	3	0.5
Expert	4	0.75
Master	5	1

In this case, the difference between the levels is 0.25 as:

$$VL_n - VL_{n-1} = 0.25.$$

In this version of the tool it is assumed that at least 5 rates are necessary to change the level so PT will be 5, and AV_j will be 0.25/5=0.05. This is therefore the value that will be added when a positive rate is received or that will be subtracted when this rate is negative. With five positive rates (5*0.05=0.25) there is thus a level promotion.

Intuition

This factor is used when the Previous Experience is low and it is necessary to use other factors to calculate a trust value. This is one contribution of our work, since most of the earlier trust models are based solely on previous experience. The agents compare their own profiles with the other agents' profiles in order to decide whether a person appears to be trustworthy or not. Therefore, the more similar the profiles of two agents are, for instance *i* and *j*, the greater the I_{ij} value in formula (1) will be. We could say that an agent 'thinks' "I do not know whether I can trust this agent but it has similar features to me so it seems trustworthy". The agents' profiles may alter according to the community in which they are working. In our case, as the data stored in the agents' profiles are 'position' and 'expertise', both these features will be taken into account. Therefore, the factors that the tool compares are:

- Experience Difference (ED)
- Position Difference (PD)

Thus, the Intuition value of an agent *i* about *j* (I_{ij}) is:

$$I_{ij} = ED_{ij} + PD_{ij} \tag{4}$$

where $ED_{ij} = LE_i - LE_j$ and $PD_{ij} = P_i - P_j$

This formula (4) is based on the idea that a person normally has a greater level of trust in people who have a higher level of experience or who are in a higher position than that person him/herself. Hence, when an agent compares its profile with another agent with higher values, the value of intuition will be positive. Let us consider the case of agent *i* which has values of $LE_i=0.75$ and $P_i=0.25$. This agent wishes to know how trustworthy another agent *j* is. In this case the agent will use Formula (1) and, depending on the information that it has about *j*, it will or will not be necessary for it to calculate the intuition factor. In this situation we shall suppose that there is little previous experience and that this must be calculated. The values for the agent *j* are $LE_j=0.25$ and $P_j=0.5$. As Figure 2 shows:

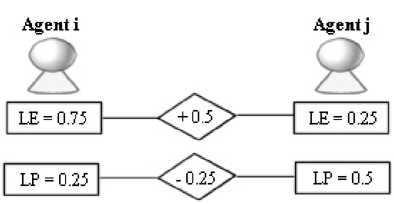

Figure 5. Comparing profiles

$I_{ij}=0.25$ as $ED_{ij}=0.5$ and $PD_{ij=}-0.25$

As with position, intuition will or will not be calculated depending on the level of PE. Thus, the weight of intuition, (see Formula 1) *wi* will be calculated as follows:
$wi = \text{int}(U/PE_{ij})$ with $PE_{ij} \neq 0$.

Previous experience

This factor is the most decisive of all the factors in Formula (1). In fact, all the previous factors depend on it as an agent will decide whether or not to use the remaining factors according to the value of Previous Experience (PE). Previous Experience is obtained through the interactions that the agent itself has, so this is direct experience. Each time one agent interacts with another (by interacting we mean that one agent uses a document provided by another), the first agent asks its user to rate that document in order to discover whether the document was: useful for him/her, related to the topic at hand, recommendable for other people interested in the same topic, up-to-date.

TABLE III. PE LABELS

Label	PE Level
Very Bad	- 0.3
Bad	- 0.2
Medium	+ 0.1
Good	+ 0.2
Very good	+ 0.3

The agent then labels this interaction with a label from Table 3. A value for Current Experience (CE) is thus obtained which will modify the previous value of PE in accordance with the following formula:

$$PE_{ij}(x) = PE_{ij}(x-1) + CE_{ij}(x) \tag{5}$$

where $PE_{ij}(x)$ is the value of Previous Experience that the agent *i* has about another agent *j* in an interaction x.

$EP_{ij}(x-1)$ is the value of Previous Experience that the agent *i* had about another agent *j* before the interaction x.

$CE_{ij}(x)$ is the value of the experience that *i* has had with *j* in the interaction x.

For instance, if an agent *i* has just taken part in an interaction with the agent *j*, and this is labeled as "bad", but the value of $PE_{ij}(x-1)$ was 0.8, then the value of $PE_{ij}(x)$ will be 0.6 obtained from (0.8+(-0.2)). Moreover the agent *i* will send the manager agent the value of $CE_{ij}(x)$ in order to calculate AV_j (see Level of Expertise).

As has previously been explained, the Position and Intuition factors depend on the PE value. When an agent has sufficient PE then Position and Intuition can be ignored, and only the PE and the LE will be considered. The latter is also included to ensure that an agent takes advantage not only of its own previous experience but also of that of the other agents since Level of Expertise (LE) is adjusted by the AV_j which comes from other previous experience.

In order to illustrate how the prototype works, let us look at an example. If a user selects a topic and wishes to search for documents related to that subject, his/her user agent will contact other user agents which have documents related to the theme at hand. The user agent will then calculate the trust value for each agent, meaning that these agents are considered to be knowledge sources and the user agent needs to calculate which "knowledge source" is more trustworthy. Once these values have been calculated, the user agent shows its user only the documents which have come from the most trustworthy agents.

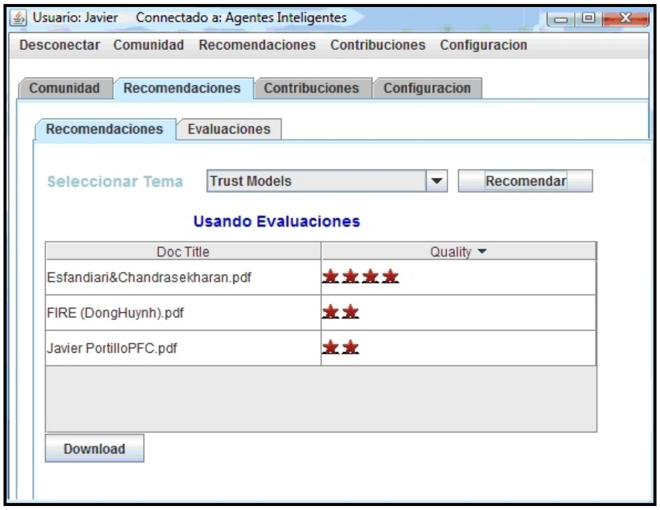

Figure 6. List of documents recommended

Figure 6 shows the results that the User Agent would display after the documents had been sorted by the trust value obtained.

V. RELATED WORK

This research can be compared with other proposals that use agents and trust models in knowledge exchange. Caballero *et al* [19] 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 an agent has about the performance of the other party to solve a given task, according to own knowledge. Abdul-Rahman & Hailes propose a model which allows agents to decide which agents' opinions they trust more and to propose a protocol based on recommendations [25]. 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 present a multi-agent belief revision algorithm based on belief networks [18]. 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 reputation is defined 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.

Huynh *et al* [15] 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 four factors and is not only a single factor.

Also, works such as Guizzardi *et al* [31] use the term 'Community' to support knowledge management but a specific trust model for communities is not used.

The main differences between these reputation models 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.

VI. CONCLUSIONS

CoPs are a means of knowledge sharing. However, the knowledge reused should be valuable for the members, otherwise CoP members might prefer to ignore the documents that a community has. In order to encourage the reuse of documents in CoPs, in this work we propose a multi-agent system to suggest trustworthy documents. Some of the advantages of our system are:

- The use of agents to represent members of the community helps members to avoid the problem of information overload since the system gives the User Agents the ability to reason about the trustworthiness of the other agents or about the recommendation of the most suitable documents to the members of the community. Users are not, therefore, flooded with all the documents that exist with regard to a particular topic, but their User Agents filter them and only recommend the most trustworthy or those which are provided by more trustworthy sources or sources which have preferences and features that are similar to them.
- Detecting whether members store documents that are not useful, since the system provides users with the opportunity to evaluate the documents consulted, and when a document is frequently evaluated with low marks then the Manager Agent will check who the provider is and whether most of that person's documents have a low evaluation. In this case, two options can be considered. First that the person does not have enough knowledge about the topic, in which case the Manager Agent can consult the Level of Expertise that this person has (which is indicated when a person

joins a community), and if this level is not suitable the Manager Agent can modify it. The second option is that this person may be consciously introducing invaluable documents. In this case the trust in this source will be low and the documents will rarely be recommended. The system can also detect the users with the greatest participation and those whose documents have obtained higher rates. This information can be used for two purposes: expert detection and/or recognition of fraudulent members who contribute with worthless documents. Both functionalities imply several advantages for any kind of organization; for instance, the former permits the identification of employee expertise and measures the quality of their contributions, and the latter permits the detection of fraud when users contribute with non-valuable information.

- The system facilitates the exchange and reuse of information, since the most suitable documents are recommended. Furthermore, this tool can be understood as a knowledge flow enabler [32], which encourage knowledge reuse in companies.

On the other hand thanks to the trust model the agents can calculate a trust value even though the community has only recently been created since, in order to calculate trust, various known factors are used such as Position, Level of Expertise and even Profile Similarity. This is a key difference with regard to other models which use only previous experience and which cannot then calculate trust values if the system is just starting to work. When a new member arrives it is also impossible for other models to calculate a previous trust value related to this new member. Moreover, the model helps to detect an increasing problem in companies or communities in which employees are rewarded if they contribute with knowledge in the community. Thus, if a person introduces, for instance, non-valuable documents with the sole aim of obtaining rewards, the situation can be detected since these documents will have lost trust values and the person will also considered to be less trustworthy. The agent will, therefore, not recommend those documents. Moreover, the formulas proposed are very simple and easy to understand. This is an advantage aver the previous models which are often not greatly used since they are difficult to implement.

Acknowledgment

An earlier version of this articles was presented at the International Conference on Information, Process, and Knowledge Management (eKNOW'09)[1].

This work is partially supported by FABRUM project. Ministerio de Ciencia e Innovación (grant PPT-430000-2008-063), PEGASO (TIN2009-13718-C02-01), MELISA (PAC08-0142-3315) and ENGLOBAS (PII2109-0147-8235) projects, Junta de Comunidades de Castilla-La Mancha, Consejería de Educación y Ciencia, in Spain, and CONACYT (México) under grant of the scholarship 206147 provided to the second author.

References

1. A. Vizcaíno, J. Portillo-Rodríguez, Soto, J.P., M. Piattini, "Encouraging the Reuse of Knowledge in Communities of Practice by Using a Trust Model", in International Conference on Information Process, and Knowledge Management (eKNOW), IEEE Computer Society, pp. 28-33, Cancun México, 2009.
2. Y. Malhotra, Knowledge Management and Virtual Organizations, IGI Publishing, Hershey, PA, USA. 2000.
3. H. Gebert, M. Geib., L. Kolbe and Riempp, G. "Knowledge-enabled Customer Relationship Management - Integrating Customer Relationship Management and Knowledge Management Concepts", in Journal of Knowledge Management, vol. 7(5), pp. 107-123, 2003.
4. E. Wenger, R. McDermott and W. Snyder, "Cultivating Communities of Practice", in Harvard Business School Press, 2002.
5. P. Hinds and C. McGrath, "Structures that work: social structure, work structure and coordination ease in geographically distributed teams", in 20th Anniversary Conference on Computer Supported Cooperative Work. 2006. Banff, Alberta, Canada.
6. H. Ushida, Y. Hirayama and H. Nakajima, "Emotion Model for Life like Agent and its Evaluation", in Proceedings of the Fifteenth National Conference on Artificial Intelligence and Tenth Innovative Applications of Artificial Intelligence Conference (AAAI/IAAI), 1998.
7. 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, pp. 152-165, 2005.
8. S. Wasserman and J. Glaskiewics, "Advances in Social Networks Analysis", Sage Publications, 1994.
9. P. Dillenbourg, "Introduction: What Do You Mean By 'Collaborative Learning'?", in Collaborative Learning Cognitive and Computational Approaches. Dillenbourg (Ed.). Elsevier Science., 1999.
10. L. Mui, A. Halberstadt and M. Mohtashemi, "Notions of Reputation in Multi-Agents Systems: A Review", in International Conference on Autonomous Agents and Multi-Agents Systems (AAMAS), pp. 280-287, 2002.
11. http://www.tripadvisor.com
12. J.P. Soto, A. Vizcaíno, J. Portillo and M. Piattini, "Applying Trust, Reputation and Intuition Aspects to Support Virtual Communities of Practice", in 11th International Conference on Knowledge-Based and Intelligence Information and Engineering Systems (KES), LNCS 4693, Springer-Verlag, pp. 353-360, 2007.
13. Y. Wang and J. Vassileva, "Trust and Reputation Model in Peer-to-Peer Networks", in Proceedings of the 3rd International Conference on Peer-to-Peer Computing, pp. 150-157, 2003.
14. B. Yu, M. Singh and K. Sycara, "An evidential model of distributed reputation management", in Proceedings of the first international joint conference on Autonomous Agents and Multiagents Systems (AAMAS), New York, USA: ACM Press, pp. 294-301, 2002.

15. T. Huynh, N. Jennings and N. Shadbolt, "FIRE: an integrated trust and reputation model for open multi-agent systems", in Proceedings of 16th European Conference on Artificial Intelligence, pp. 18-22, 2004.
16. J. Sabater and C. Sierra, “Reputation and Social Network Analysis in Multi-Agent Systems”, in proceedings of the first international joint conference on autonomous agents and multiagent systems (AAMAS), New York, USA: ACM Press, pp. 475-482, 2002.
17. W. Taecy, G. Chalkiadakis, A. Rogers and N. Jennings, “Sequential Decision Making with Untrustworthy Service Providers”, in Proceedings of 7th International Conference on autonomous Agents and Multiagent Systems (AAMAS), pp. 755-762, 2008.
18. K. Barber and J. Kim, "Belief Revision Process Based on Trust: Simulation Experiments", in 4th Workshop on Deception, Fraud and Trust in Agent Societies, pp. 1-12, 2004.
19. A. Caballero, J. Botia 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., André, E., Zhong, N. (eds), LNCS 4196, pp. 172-183, 2006.
20. R. Hermoso, H. Billhardt and S. Ossowski, “Integrating trust in virtual organizations”, in International Workshop on Coordination, Organization, Institutions and Norms in Agents Systems II: AAMAS nad ECAI, COIN 2006, LNAI, pp. 19-31, 2007.
21. J. Carter, E. Bitting, and A. Ghorbani, “Reputation for an information-sharing multi-agent system”, in Computational Intelligence, vol. 18, no. 2, pp. 515-534, 2002.
22. L. Mui, M. Mohtashemi, and A. Halberstadt, “A Computational Model of Trust and Reputation for E-businesses”, in Proceedings of the 35th Hawaii International Conference on Systems Sciences (HICSS), IEEE Computer Society Press, 2002.
23. A. Jøsang, R. Ismail, and C. Boyd, “A Survey of Trust and Reputation Systems for Online Services Provision”, in Decision Support Systems, vol. 43, no. 2, pp. 618-644, 2007.
24. J. Sabater and C. Sierra, “*Review on Computational Trust and Reputation Models*”, in Artificial Intelligence Review, vol. 24, pp. 33-60, 2005.
25. A. Abdul-Rahman and S. Hailes, "Supporting Trust in Virtual Communities", in Proceedings of 33rd Hawaii International Conference on Systems Sciences (HICSS'00), IEEE Computer Society, vol. 6, pp. 1769-1777, 2000.
26. B. Misztal, "Trust in Modern Societies", Polity Press, Cambridge MA, 1996.
27. I. Nonaka and H. Takeuchi, "The Knowledge Creation Company: How Japanese Companies Create the Dynamics of Innovation", Oxford University Press, 1995.
28. R. Hardin, "The Street Level Epistemology of Trust", in Politics and Society, vol. 21, pp. 505-531, 1993.
29. A. Jøsang, "The Right Type of Trust fro Distributed Systems", in New Security Paradigms, 1996.
30. N. Luhmann, "Trust and Power", in Wiley, Chichester, 1979.
31. R. Guizzardi-Silva, L. Aroyo and G. Wagner, "Help&Learn: A Peer-to-Peer Architecture to Support Knowledge Management in Collaborative Learning Communities" in Revista Brasileira de Informatica na Educação, vol. 12(1), pp.29-36, 2003.
32. O. Rodríguez-Elias, A. Martínez-García, A. Vizcaíno, J. Favela and M. Piattini, "A Framework to Analyze Information Systems as Knowledge Flow Facilitators", Information Software Technology, vol. 50(6), pp. 481-498, 2007.
33. J.P. Soto, A. Vizcaíno, J. Portillo-Rodríguez, and M. Piattini, “Why Should I Trust in a Virtual Community Member?”, in 15th Collaboration Researchers’ International Workshop on Groupware (CRIWG), LNCS 5784, pp. 126-133, 2009.