USING AGENTS TO CONFRONT SOME OF THE CHALLENGES OF KNOWLEDGE MANAGEMENT SYSTEMS

Javier Portillo-Rodríguez[1], Aurora Vizcaíno[1], Juan Pablo Soto[2] and Mario Piattini[1]
[1]ALARCOS Research Group, Information Systems and Technologies Department
UCLM-INDRA Research and Development Institute, Escuela de Informática, Universidad de Castilla-La Mancha
Paseo de la Universidad 4 - 13071 Ciudad Real, Spain

[2]Mathematics Department, University of Sonora
Blvd. Luis Encinas y Rosales s/n, Col. Centro, CP. 83000, Hermosillo, México

Keywords: Agent architecture, Knowledge management systems, Communities of practice, Trust models.

Abstract: The importance of knowledge management has, in recent years, led to the incorporation of Knowledge Management Systems (KMS) into companies. Some of these KMS could be considered as Recommender Systems that are able to recommend knowledge, which is part of the company's intellectual capital. However, these KMS are not always welcome in the company, since the knowledge is not stored by using a quality control, or because employees feel that these kinds of systems, rather then helping them, cause them extra work. In this paper we present an agent architecture combined with a trust algorithm trying to avoid some of the problems that appear when a KMS is introduced into companies.

1 INTRODUCTION

In recent years, knowledge has become an extremely important factor (Hansen and Kautz, 2004). Subjects such as Knowledge Management are, therefore, currently of particular interest to organizations who are concerned about their employees' learning and competitiveness, since a suitable management of knowledge can help them to increase their members' collaboration and encourage them to share knowledge. At present organizations must operate in a climate of rapid market change and high information volume, which increases the necessity to create knowledge management systems (KMS) that support the knowledge process. It is possible to consider certain Recommender Systems as KMS, however, these kinds of systems are not always welcomed by a company's employees because (Lawton, 2001) on occasions employees waste a considerable amount of time searching for information, with regard to this, sometimes there is no quality control with regard to the KOs (Knowledge Objects) introduced into the system and employees may introduce information into the systems which is not very valuable.

Our work is focused on attempting to reduce the impact of these problems. We therefore use software agents to search for information on behalf of users, and these agents are in charge of recommending the most suitable knowledge to them.

We pretend to use our proposal in Communities of Practice (CoPs) which are a natural means of sharing knowledge, which is considered to be a critical factor for an organization's competitive advantage (Hansen and Kautz, 2004).

However, nowadays, these kind of communities, due to globalization, are geographically distributed and there are no face-to-face interactions. If CoP members are distributed and they do not know the other members trust between CoP members decrease. This situation could be a problem because people in general prefer to exchange knowledge with "trustworthy people" and if there is not enough trust among members knowledge exchange could decrease too. People with a consistently low reputation will eventually be isolated from the community since others will rarely accept their justifications or arguments and will limit their interactions with them. This issue, plus the problems pointed out previously, have led us to develop an agent architecture and a recommendation algorithm

Portillo-Rodríguez J., Vizcaíno A., Pablo Soto J. and Piattini M. (2010).
USING AGENTS TO CONFRONT SOME OF THE CHALLENGES OF KNOWLEDGE MANAGEMENT SYSTEMS.
In *Proceedings of the 2nd International Conference on Agents and Artificial Intelligence - Agents*, pages 249-252
DOI: 10.5220/0002699202490252

to encourage the reuse of knowledge in CoPs. In order to tackle these problems, we have developed an agent architecture and a trust algorithm with which to rate KOs and Knowledge Sources (KSs) that produce these KOs. The software agents will therefore use this algorithm in order to decide whether a KO or KS should be recommended to a particular user.

Therefore in Section 2 the agent architecture is described and later, in Section 3, a recommender system and the recommender algorithm used by this system is explained. . Finally, our conclusions are outlined in Section 4.

2 AN AGENT ARCHITECTURE

The agent architecture proposed is composed of two levels: reactive and deliberative-social. The reactive level is considered by other authors to be a typical level that an Agent Architecture must have (Ushida, 1998). A deliberative level is often also considered as a typical level, but a social level is not often considered in an explicit manner, 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 behaviour. Since we wish to emulate human feelings such as trust when working in CoPs, we have added a social-deliberative level that considers the social aspects of a community and which takes into account the opinions and behaviour of each of the members of that community.

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

2.1 Reactive Level

This is the level in charge of perceiving changes in its environment and responding to these changes at the precise moment at which they occur, i.e., when an agent executes another agent's request without any type of reasoning.

The components of the reactive level are (see Figure 1):

Internal Model. This component stores the individuals' features. These features will be consulted by other agents in order to discover more about the person represented by the User Agent

Beliefs. This module is composed of inherited beliefs (pre-defined beliefs) and lessons learned (obtained by interaction with the environment) from the agent itself.

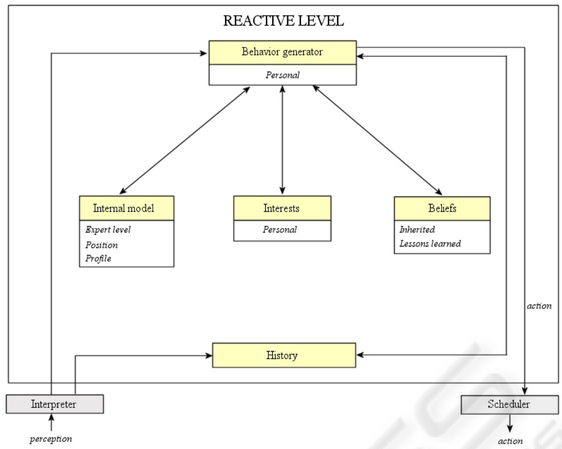

Figure 1: Reactive Level.

Interests. These are a special kind of beliefs. This component represents individual interests that an agent has with regard to a topic or a knowledge source.

Behaviour Generator. This component is fundamental to our architecture. It is here that the actions to be executed by the agent are triggered. Depending on the information received from the *Interpeter* module the agent makes a matching process to select the correspondent behaviour.

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 2):

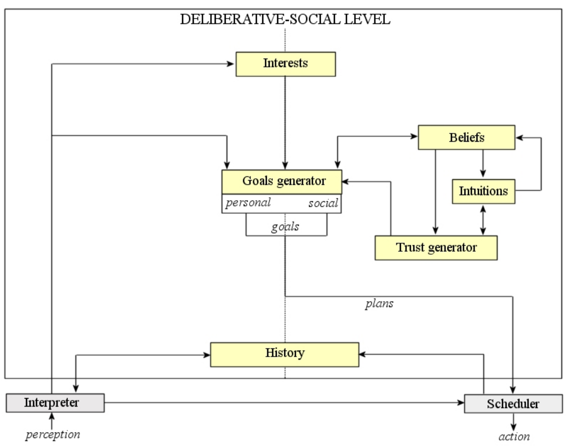

Figure 2: Deliberative-Social Level.

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 represents interest in other agents.

Intuitions. We often trust more in people who have similar features to our own. Thus, 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. The agents' profiles may change according to the community in which they are working. This factor will be used in those cases when the agent doesn't have enough information to know if a KS is trustworthy.

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

Trust Generator. This module is in charge of generating a trust value for the knowledge sources with which an agent interacts in the community. To do this, the trust generator module considers the trust model explained in detail in (Soto et al, 2007) which considers the information obtained from the internal model and the agent's intuitions.

3 A RECOMMENDER SYSTEM

A recommender system has been developed in order to test the trust model and the multi-agent architecture. In this system each CoP member is represented by a software agent called a User Agent. A new community member must first join a community, which is done by using the "*Register"* menu and choosing a community from those which are available. Once registered, a member can provide new KOs or use those which are already available in the community and/or propose new subjects. One way to obtain KOs in a community is requesting a KO recommendation. To obtain a KO recommendation user has to use the "Recommend" menu and select a topic. To make the recommendation, the prototype will use a recommendation algorithm that has been design as follows.

The input the algorithm is a set of KOs. Each KO may or may not have been evaluated previously, signifying that a KO may already have a list of evaluations (along with the identity of each person who evaluated it), or it may not have any evaluations. This aspect will be taken into account by the algorithm, which therefore distinguishes two groups:

Group 1 (G1): This group is formed of the KOs that have already been evaluated. This is the most important group since if the agents have previous evaluations of a KO they have more information about it, which facilitates the task of discovering whether or not its recommendation is advisable.

Group 2 (G2): these KOs have not been used previously so the agents do not have any previous evaluations of them. Let us now observe how each group is processed by the algorithm.

In G1 the KOs will be ordered by a Recommendation Rate which is calculated by the User Agent for each KO. Hence RR_k signifies the Recommendation Rate for a particular KO called k, and is obtained from:

$$RR_k = w1 * TE_i + w2 * TS_{ik} \tag{1}$$

where TE_i is the pondered mean of the evaluations determined by the trust that an agent "i" has in each evaluator (the person who has previously evaluated that KO). TE_i is calculated as:

$$TE_i = \frac{\sum_{j=1}^{n} E_{jk} * TS_{ij}}{\sum_{j=1}^{n} TS_{ij}} \tag{2}$$

Therefore, TS_{ij} is the trust value that the User Agent "*i*" has in the knowledge source "*j*", since in a CoP the source which provides a KO will usually be a CoP member. TS_{ij} therefore represents the trust that an agent "*i*" has in another agent "*j*" and E_{jk} is the evaluation that an agent "*j*" has made with regard to a particular KO "*k*".

The parameter TS_{ik} used in Formula (1) similarly indicates the trust that an agent "*i*" has in a knowledge source "*k*". In other words, the agent must take two things into consideration when calculating the RR_K

- The other agents' opinions of a KO "k" pondered by the trust that agent "i" has in the person who provided that evaluation.
- The opinion that the agent "i" has in the agent that has provided the KO "k".

Both w1 and w2 are weights which are used to adjust the formula. The sum of w1 and w2 should be 1.

Group 2 will use another formula to calculate the RR_k for each KO since, in this case, there are no results of previous evaluations of the KOs. This formula, not explained due to space problems, basically uses a pondered mean of the trust values that other agents have about the KS.

4 CONCLUSIONS

CoPs are a means of knowledge sharing. However, the knowledge that is reused should be valuable for its members, who might otherwise prefer to ignore the documents that a community has at its disposal. In order to encourage the reuse of documents in CoPs, in this work we propose a multi-agent recommender system with which 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 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 subject, but their agents filter them and recommend only those which are most trustworthy (when they have rates) or those which are provided by more trustworthy sources or sources which have preferences and features that are similar to those of the user in question.
- The system can detect those users with the greatest level of 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 various advantages for any kind of organization, i.e., 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. The tool can also be understood as a knowledge flow enabler (Rodríguez-Elias et al, 2007), which encourages knowledge reuse in companies.

Furthermore, the proposed algorithm is quite flexible since in many situations weights are used to modify the formulas. This algorithm could, therefore, be used by the designers of other recommender systems who could decide what values they should give to these weights in order to adapt the formula to their needs.

REFERENCES

Hansen, B. and Kautz, K., (2004), "Knowledge Mapping: A Technique for Identifying Knowledge Flows in Software Organisations", in Software Process Improvement, LNCS 3281, Springer: 126-137.

Lawton, G., (2001), "Knowledge Management: Ready for Prime Time?", in Computer, vol. 34(2): 12-14.

Rodríguez-Elias, O., Martínez-García, A., Vizcaíno, A., Favela, J., and Piattini, M. (2007). "A Framework to Analyze Information Systems as Knowledge Flow Facilitators", in Information Software Technology, vol. 50(6): 481-498.

Soto, J. P., Vizcaíno, A., Portillo, J., and Piattini, M., (2007), "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: 353-360.

Ushida, H., Hirayama, Y., and Nakajima, H., (1998), "Emotion Model for Life like Agent and its Evaluation", in Proceedings of the Fifteenth National Conference on Artificial Intelligence and Tenth Innovative Applications of Artificial Intelligence Conference (AAAI'98 / IAAI'98), Madison, Wisconsin, USA.