A Tool with which to Recommend Knowledge Sources in Virtual Communities

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

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

Abstract: This paper describes a multi-agent architecture with which to recommend knowledge sources in virtual communities. The tool presented in this work may be particularly useful to software organizations for two main reasons: it can assist them in the management, generation, acquisition, exchange, protection, distribution, and utilization of knowledge generated during the development of their projects, and it can also help to increment the collaboration of their members by, for example, supporting the sharing and use of knowledge between them. The applicability of the tool is illustrated with a scenario in order to show its features.

1 Introduction

Knowledge Management (KM) provides methods and techniques that can help organizations to increment the collaboration of their members by, for example, supporting the sharing of knowledge between them. Documented examples of benefits that may be attained from managing knowledge effectively include: reduced time-to-market; reduced development costs; innovative uses of existing products; revolutionary product ideas; and reduced employee turnover [Sk03]. An essential ingredient of knowledge sharing in organizations is that of virtual communities, by which we mean groups of people whose members may or may not meet one another face to face and may exchange word and ideas through the use of computer networks [GBK04]. This concept has become more and more popular within the field of the KM where it is mainly used as a KM tool to support the externalization of knowledge, both for reuse and for the purpose of innovation [HW00]. The importance of the concept of a virtual community at an organizational level is parallel to the growth in the interest of management approaches such as organizational learning and knowledge management. Virtual communities are often used in software development groups since software development is a knowledge intensive work in which software engineers must collaborate with other members of the team, and share their knowledge and experience in order to complete their assignments.

We believe that by providing software development groups with tools that facilitate their members' collaboration and sharing of knowledge, the performance of these communities can be incremented. In order to do this, we have designed a recommendation tool with which to identify knowledge sources and provide useful knowledge to virtual community members. The remainder of this work is organized as follows. The next section defines the concept of the virtual community and its advantages in software organizations. In Section 3 the multi-agent architecture used to develop our tool is described. Section 4 illustrates the applicability of the tool. In Section 5 related works are outlined. Finally, in Section 6 conclusions are presented.

2 Virtual communities

Formal and informal networks play an increasing role in business life and are a fundamental part of learning and knowledge exchange. Whilst the social aspect of learning and knowledge sharing plays a key role in these types of networks, the need to network and share knowledge through a chain of interdependent organizations (globalization) means that members of such networks are often not located at the same site or even in the same country. This has led to the emergence of virtual communities through which to keep the members of these networks in touch even without face-to-face meetings. According to Hamman, the sociological term 'community' should be understood as meaning a group of people who share social interaction, and some common ties between themselves and other members of the group, and who share an area for at least some of the time [Ha01].

In contrast to the sociological definitions of 'community' in which place and physical presence are important aspects, it is clear that computer networks allow for communities that stretch well beyond the neighbourhood [WG99]. In summary then, virtual communities are seen as the intersection of social and technical systems [SS01], and, as is recognized by [MAI97], neither technology nor sociality can supplant the need for the other. On the other hand, the use of virtual communities is increasing day by day. Virtual community members may or may not meet one another face to face and may exchange words and ideas through the use of computers networks [GBK04]. Consequently, the association of community members and the enabling electronic medium constitute an interesting infrastructure that provides benefits by supporting interpersonal relationships and companionship, encouraging discussion and knowledge sharing, allowing for quick access to information and enabling collective action such as, for instance, software development [Bu01]. Our research is focused upon professionally-orientated communities (software organizations), which consist of company employees who communicate and share information in order to support their professional tasks.

Software development communities

Many changes have recently taken place in the software development process, and software development is, to a large extent, distributed in that developers and users are geographically dispersed. Multi-site development is a current matter of study and discussion, since global development is becoming a usual style of software production. Offshore and nearshore outsourcing are practices which are therefore increasing considerably. Moreover, when participants are distributed geographically new problems often arise. For example, communication and coordination are more difficult because of differences in culture, timetable, language, etc., [BF04]. Some areas of research exist which attempt to minimize the impact of these problems. One such area is that of communities. These groups in software organizations provide the necessary infrastructure through which to exchange information between software engineers. The adequate management of these communities provides the necessary information to respond to certain questions that must be answered, such as: What kinds of problems could be solved in the community? What is the knowledge involved in the activities performed by the software developing groups? How do they share that knowledge? and who are the expert employees in topics related to the community? For these reasons, we consider it important to implement tools which give support to virtual communities in software organizations with the goal of recommending knowledge sources (which may be documents or ever community members). Before describing the proposed tool in detail, we shall first describe the architecture used to develop it.

3 Multi-agent architecture

The goal of this work is to provide an infrastructure for KM in virtual communities. In order to carry this out, we first designed a multi-agent architecture in which software agents try to emulate the behaviour of the employees in communities. This architecture provides the capacity to:

- Assist employees in identifying trustworthy entities and provide the confidence necessary to foster the usage of information and knowledge of recommendation systems. In order to do this, we have designed a reputation formula based on real world social properties (expertise, previous experience, intuition and position) of trust in communities [SVP07]. The main goal of this formula (which will be explained in detail in Section 4) is to rate the level of confidence in an information source or in a provider of knowledge. This formula could also help companies to detect those employees with more knowledge about a topic (expert detection).
- Give artificial agents the ability to reason about the trustworthiness of the agents or of a knowledge source in virtual communities.
- Encourage knowledge exchange between the community members.

Taking these facts into account, we designed a multi-agent architecture which is composed of two levels (see Figure 1): the reactive and the deliberative-social level.

The reactive level is considered by other authors as a typical level that a multi-agent system must have [UHN98]. 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 prepared to represent this or other behaviour. In our case the deliberative and the social level are not separate levels because, after developing several prototypes, we realised that in our domain both are narrowly related.

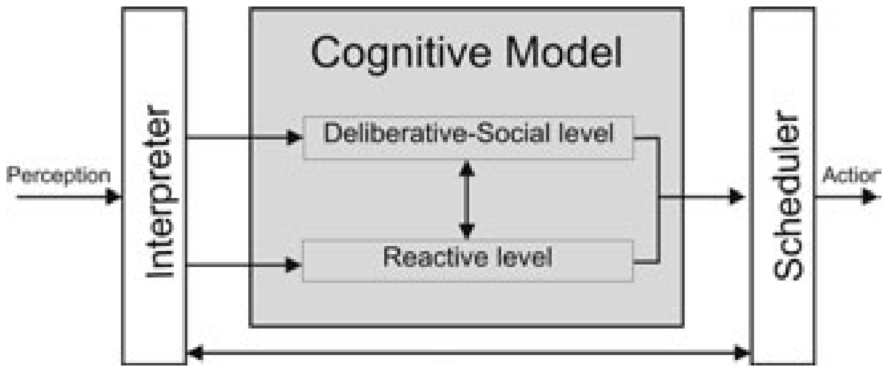

Figure 1: General architecture

Reactive level: In this level we represent 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: In this level the agent's behaviour is based on goals, that is, the agent has several defined goals and it attempts to achieve these goals by scheduling plans. Due to the fact that we are trying to represent human behaviour in a virtual community, 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. It is for this reason that we have considered both the social and deliberative aspects. The former attempts to achieve individual goals and the latter is more focused upon achieving social goals (community 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 best suited to the achievement of its goals.

Two further important components of our architecture are an *Interpreter* and a *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. Due to space constrains the modules that form each of the levels have been omitted.

4 Recommendation tool

In order to test our proposal 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 software development groups to discover 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 software organizations. In addition, we attempt to avoid the situation of software engineers 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 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 implemented 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 behaviour 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.
- The person uses knowledge previously stored in the community.
- The person helps other users to achieve their goals, for instance by giving an evaluation of certain knowledge.

The second type of agent within a community is called the *Manager Agent* (which must manage and control its community). In the following paragraphs we shall explain how the prototype works.

Firstly, when a user wants to join a community in which no member knows anything about him/her, the reputation value assigned to the user in the new community is calculated on the basis of the reputation assigned from others communities in which the user is or was a member. For instance, a User Agent called *j*, will ask the managers of each community in which s/he was previously a member to consult each agent that knows him/her with the goal of calculating the average value of his/her reputation (R_{Aj}). This is calculated as:

$$R_{A_j} = (\sum_{j=1}^{n} R_{s_j})/n$$

where n is the number agents who know j and $\mathbf{R}_{sj}$ is the value of reputation of j in the eyes of s. In the case of being known in several communities the average of the values $\mathbf{R}_{Aj}$ will be calculated. The User Agent j then presents this reputation value (similar to when a person presents his/her curriculum vitae when s/he wishes to join a company) to the Manager Agent of the community to which it is "applying". This reputation value permits an initial reputation value to be assigned, taking into account the previous experiences and relations with others agents, thus generating a flow and exchange of information between the agents. This mechanism is similar to the "word-of-mouth" propagation of information for a human [AH00].

In addition, $\mathbf{R}_{sj}$ value is computed as follows:

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

where R_{sj} denotes the reputation value that $agent_s$ has in $agent_j$ (each agent in the community has an opinion about each of the other agent members of the community with which it has interacted). E_j is the value of expertise which is calculated according to the degree of experience that a person has in a domain and which is given by the company. P_j is the value assigned to a person's position. This position is defined by the enterprise's organizational diagram. Therefore, a value that determines the hierarchic level within the organization can be assigned to each level of the diagram. I_{sj} denotes the intuition value that $agent_s$ has in $agent_j$ which is calculated by comparing each of the users' profiles.

In addition, previous experience should also be calculated. We suppose that when an agent *s* consults information from another agent *j*, the agent *s* should evaluate how useful this information is. This value is called QC_{sj} (Quality of *j*'s Contribution). To attain the average value of an agent's contribution, we calculate the sum of all the values assigned to their 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 Reputation value can be adjusted to the needs of the organizations. For instance, if an enterprise considers that all its employees have the same category, then $w_p = 0$. The same could occur when the organization does not take its employees' intuitions or expertise into account. In this way, an agent can obtain a value related to the reputation of another agent and decide to what degree it is going to consider the information obtained from this agent.

Finally, when a person searches for a document relating to a topic his/her User Agent consults the Manager Agent about which documents are related to its search. Then, the Manager agent answers with a list of documents. The User Agent sorts this list according to the reputation value of the authors, which is to say that the contributions with the best reputations for this Agent are listed first. On the other hand, when the user does not know the contributor then the User Agent consults the Manager Agent about which members of the community know the contributors. Therefore, the User Agent can consult the opinions that other agents have about these contributors, thus taking advantage of other agents' experience. To do this the Manager consults its interaction table and responds with a list of the members who know the User Agent This User Agent then contacts each of them. If nobody knows the contributors then the information is listed, taking their expertise and positions into account. In this way the User Agent can detect how worthy a document is and recommend it, thus saving employees' time, since they do not need to review all the documents related to a topic but only those considered most relevant by the members of the community or by the person him/herself according to previous experience with the document or its authors.

5 Related work

This research can be compared with other proposals that use agents and trust in knowledge exchange. For instance, in [AH00], the authors propose a model that allows agents to decide which agents' opinions they trust more and 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 keep rather complex data structures that represent a kind of global knowledge about the whole network.

Barber and Kim present a multi-agent belief revision algorithm based on belief networks [BK04]. 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, the source should indicate the certainty factor that the source has of that knowledge. In our case, the focus is very different since it is the receiver who evaluates the relevance of a piece of knowledge rather than the provider as in Barber and Kim's proposal.

6 Conclusions

The main contribution of this paper is the design of a recommendation tool with which to identify and recommend knowledge sources in virtual communities. The advantages of this contribution are that:

- It improves the management of knowledge in software organizations. This implies several advantages for organizations since it encourages groups of employees to exchange information. It is therefore expected that a greater flow of communication will exist between them which will consequently produce an increase in their knowledge.
- It gives information about the location of information, best practices and expertise.

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

Acknowledgements

This work is partially supported by the MECENAS (PBI06-0024), ESFINGE (TIN2006-15175-C05-05) and MELISA project (PAC08-0142-3315), 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 first author.

References

[AH00] Abdul-Rahman, A., Hailes, S.: Supporting Trust in Virtual Communities. Proceedings of the 33rd Hawaii International Conference on Systems Sciences (HICSS'00), Vol. 6, 2000.

[BK04] Barber, K., Kim, J.: Belief Revision Process Based on Trust: Simulation Experiments. 4th Workshop on Deception, Fraud and Trust in Agent Societies, Montreal Canada, 2004; pp. 1-12.

[BF04] Boland, D., Fitzgerald, B.: Transitioning from a Co-Located to a Globally-Distributed Software Development Team: A Case Study at Analog Devices Inc. 3rd. International Workshop on Global Software Development, Edinburgh, Scotland, 2004; pp. 4-7.

[Bu01] Butler, B.: Membership Size, Communication Activity and Sustainability: A Resource-based Model of Online Social Structures. Information Systems Research, Vol. 12, No. 4, 2001; pp. 346-362.

[GBK04]Geib, M., Braun, C., Kolbe, L., Brenner, W.: Measuring the Utilization of Collaboration Technology for Knowledge Development and Exchange in Virtual Communities. Proceedings of the 37th Annual Hawaii International Conference on Systems Sciences (HICSS), Vol. 1, 2004.

[Ha01] Hamman, R.: Computer Networks Linking Networks Communities. Werry, C., and Mowbray, M. (Eds.), Online Communities, Prentice Hall, 2001; pp. 72-92.

[HW00] Huysman, M., Wit, D.: Knowledge Sharing in Practice. Dordrecht, 2000.

[MAI97] Mynatt, A., Addler, A., Ito, M., O'Day, V.: Design for Network Communities. Online Proceedings of CHI (www.acm.org(sigs/chi97/proceedings), 1997.

[Sk03] Skyrme, D.: Knowledge Management: Making Sense of an Oxymoron. Management Insight, Vol. 22, 2003.

[SVP07] Soto, J. P., Vizcaíno, A., Portillo, J., Piattini, M.: Applying Trust, Reputation and Intuition Aspects to Support Virtual Communities of Practice. 11th International Conference on Knowledge-Based and Intelligence Information & Engineering Systems (KES), LNCS 4693, pp. 353-360, 2007.

[SS01] Stanoevska-Slabeva, K., Schmid, B. (2001). "A typology of online communities and community supporting platforms." Proceedings of the 34th Hawaii International conference on System Sciences (HICSS) **7**: 7010.

[UHN98] Ushida, H., Hirayama, Y., Nakajima, H.: 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'98 / IAAI'98), Madison, Wisconsin, USA, 1998: pp. 8-37.

[WG99] Wellman, B., Gulia, M.: Virtual Communities as Communities: Net Surfers Don't Ride Alone. Smith, M.A., and Kollock, P. (Eds.), Communities in Cyberspace. Routledge, London, 1999.