A Flexible and Active Community for Supporting Collaborative Learning

Xu Chen[1], Xiaojun Cheng[2], and Zhiyong Peng[3]

[1]State Key Lab of Software Engineering,Wuhan University,Wuhan,430072,China
xu_chen@mail.whu.edu.cn
[2]State Key Lab of Software Engineering,Wuhan University,Wuhan,430072,China
csforgood@gmail.com
[3]Computer School, Wuhan Universtiy, Wuhan,430072, China
peng@whu.edu.cn

Abstract. Collaborative learning in a community would be better than students studying individually. Inspiring learners' interests and supporting community evolution is essential for a learning community. We propose a flexible and active community supporting the collaborative learning. This community takes into full consideration of the dynamic nature of community and provides a user-interactive question and answer (QA) system for knowledge management. This paper introduces a flexible community management in support of community division and merger, and a QA system together with forestalling-answer contest system for efficiently collaborative learning.

Keywords: Collaborative Learning, Community

1 Introduction

With the advance of Internet in the 1990s, Computer Supported Collaborative Learning (CSCL) becomes more and more important. It proposes the development of new software and applications that bring learners together and that can offer creative activities of intellectual exploration and social interaction [1]. Instead of simply posting the leaning material as traditional anti-social education, CSCL stresses learning through co-construction and personal reflection, which means students acquire knowledge by expressing their questions, helping other people to obtain answers and share ideas or learning materials. In addition, the instructors are not only required to prepare materials and make them available by computer, but also perform a minimal pedagogical intervention in order to redirect the group work in a productive direction [2].

The shift to the group unit of analysis coincided with a focus on the community as the agent of situated learning [1]. In 1991, researchers focus on the "community of practice (CoP)"[3],firstly created by Lave, J. and Wenger E., as a vehicle for learning, because learning is a process of becoming a member of a sustained community of practice, rather than a process of socially shared cognition that results in the end in the internalization of knowledge by individuals[4]. Their examples were initially all

Joseph Fong, Reggie Kwan, Fu Lee Wang (Eds.): Hybrid Learning: A New Frontier, pp. 11-21.
ICHL 2008, City University of Hong Kong, Hong Kong, August 2008.

apprenticeship based as their theory of learning through a process called Legitimate Peripheral Participation (LPP) [3]. During the mid-1990s, CSCL began to explore how computers could bring students together to learn in the smaller community. Because the shared construction of meaning is most visible and available for research at the small-group unit of analysis. Moreover, the knowledge building that takes place within small groups becomes "internalized by their members as individual learning and externalized in their communities as certifiable knowledge" [5]. Actually, participation in a community operation is always operating with diverse concept of what CSCL is all about. Sfard defines two broad and irreconcilable metaphors of learning that are necessarily relevant to CSCL: the acquisition metaphor, in which learning consists of individuals acquiring knowledge stored in their minds, and the participation metaphor, in which learning consists of increasing participation in communities of practice [6].

Good community design requires an insider's perspective to lead the discovery of what the community is about [7]. According to their participation levels, there are different roles of community members. (1) Potential members: those who are not members of the community but who are interested in the community. (2) Peripheral members: a large portion of community participators, who remain peripheral and make little contribution to the community. (3) Active members: those who are active in the community and attend meeting regularly. (4) Leaders: the small core group who lead and organize events and manage the community. By participating frequently and collaborating in the community, the latter two groups' members can learn better than the former two groups. Only 25 to 35 percent of members belong to these two groups. Active members may be deeply engaged for some time and then leave the community. Because some may feel that their interests have changed or the community has moved in a direction that they don't agree with. Some may gradually lose interests because there is no incentive for them to participate or collaborate in the community. Due to the dynamic nature of the community, the role of a community member would change. For example, those peripheral members would drift to center and become quite involved as they observe some interesting activities in the community.

Therefore, inspiring learners' interests and supporting community evolution are essential for development of a learning community. This paper proposes a flexible and active community to support collaborative learning. The flexible community management takes into full consideration community leaders' ever-changing needs for community division and merger as well as individual learners' needs for finding community and playing different roles in different communities. And the active learning community integrates a user-interactive QA system with a forestalling-answering contest system.

After having introduced a motivating example in section 2, the paper would present a flexible community design based on Object Deputy Model [8] in section 3, followed by a general design of a user-interactive QA system in section 4, and then a forestalling-answer system in section 5.And the conclusions will be presented in section 6.

2 Motivation

Before we move on to give a formal definition of our community, this section gives a motivating example to illustrate the necessities and benefits of a dynamic and active community.

Tom is a new master in a university. He is specialized and interested in programming. Thus, he would like to participate in the programming community. But he does not know which community can provide programming services if there is not any community searching mechanism. After the appropriate one is found eventually, he may be gradually involved in the community. However, his passion would be weaken if he carries litter authority even if he makes great contribution or has great knowledge of programming. Besides, he may find his really interest is Java, but there is not a Java community currently. And then if some one would create a Java community afterwards, he will be wondering if he could attend both two communities but play different roles. Several problems exist in this scenario includes:

(1) How to find the appropriate community you are interested in?
(2) If the right community does not exist, how to create a new one?
(3) How to keep members' passion in a community?
(4) How to play different roles in different communities?

A flexible and active community can provide efficient support for Tom. A community recommendation mechanism would take Tom to his favourite community, and a flexible community management would provide an easy method to create, divide and merge communities. In addition, those communities based on Object Deputy Model could also provide multiple roles for the same member in different communities. A reputation-based QA system together with a forestalling-answer contest system would keep member active in the collaborative learning community.

3 A flexible Community

Data management of the community is on the basis of Object Deputy Model[8], which can provide more flexibility than traditional object-oriented data model. It can reflect the dynamic nature of the community, and allow a user to play multiple roles in different communities and recommend communities.

3.1 Community Definition

A community is defined as a deputy class[8] consisting of five attributes.

Community=<name, members, services, constrains, domain knowledge>

For modeling purpose, a community is defined as a set of name, members, services, constrains and domain knowledge. Members and domain knowledge are the optional attributes while the other three are mandatory. Name is used to identify the community. Services are used to support collaborative learning. For example, a chat room can be used as an instant text-based communication tool. The constraint is the regulation for users' participation. For instance, persons who are interested in

programming and have at least a bachelor degree can be members of an exclusive programming community. Domain knowledge is helpful for community members to learn and is usually generated from QA system described in the section 4. The members are the users which is a source class[8] defined as follows,

User=< static attributes, dynamic attributes, additional attributes >

The source class “users” has three kinds of attributes, namely static attributes, dynamic attributes and additional attributes. Static attributes represent a set of static information filled in the registration, such as name, gender, birthday and original interests. The dynamic attributes represent one member’s community dynamic status, such as the user reputation and the social role. The additional attributes are designed for community evolution, such as an extensible attribute for a new community.

3.2 Community Flexibility

The main advantage of a flexible community is that it can be easily divided and merged.

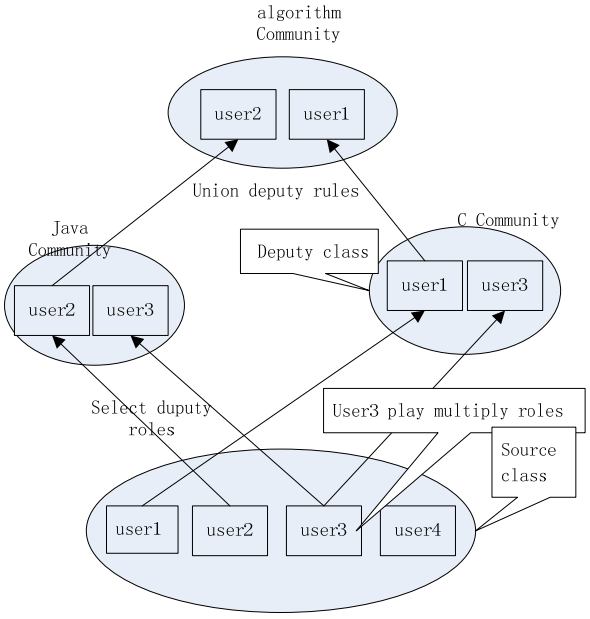

Fig. 1. Community Management Overview

As shown in Figure 1, there are three kinds of operations for the community management (1) A community is created by defining a deputy class of users which is the source class. In Figure 1, there are four source objects in the source class, namely user1, user2, user3 and user4. Both user2 and user3 are interested in Java while both user1 and user3 are interested in C. According to the predefined selection operation[8], user2 and user3 are classified into Java community, and user1 and user3 into C community. As for user3, he can be members of both communities but play different roles in these two communities, because he is interested in both Java and C

knowledge. (2) Two communities can be merged by the union operation. For example, both user1 and user2 are interested in algorithm and the algorithm community can be created by merging Java community and C community so that user1 and user2 can be its members. (3) A community can be divided into several communities by the selection operation. For example, the user community can be divided into Java community and C community.

3.3 Community Recommendation

We propose a community recommendation mechanism for users' participation. For example, Tom can be recommended with some communities he may be interested in by taking the following five steps.
Step 1: Tom firstly registers in and fills in the basic information which includes name, age, gender, interests, education background and so forth.
Step 2: Because Tom is interested in programming and has a bachelor degree, he satisfies the selection condition of the programming community which will be automatically recommended to TOM.
Step 3: After TOM joined the community, he can find his friends or ask some questions in the community.
Step 4: Based on the semantic relationship among communities, a community family tree can be shown in Figure 2.

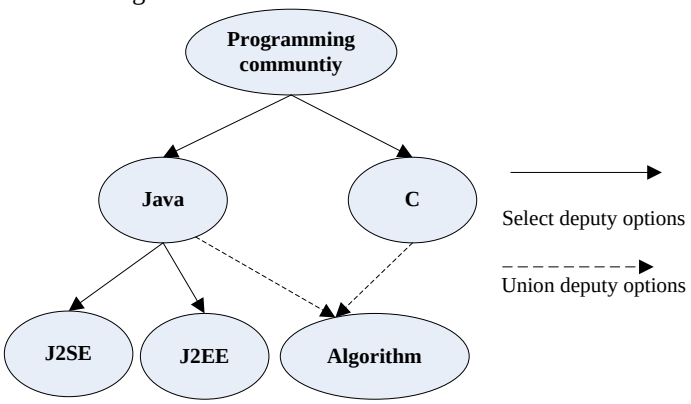

Fig. 2. A Community Family Tree

Step 5: Tom may join Java community if he is more interested in Java than C while still remains the membership of the programming community.

4 User-interactive QA system

We design a user-interactive QA system together with the user reputation model. During collaborative learning in the community, students are encouraged to ask questions, and knowledge is a product out of interaction. Members acquire knowledge

by asking question and answering others' questions. Additionally, they would earn a high reputation if they could provide more and better answers.

Unlike the traditional automatic QA system, a user-interactive QA system serves as an interactive platform for users to help each other by answering questions, which overcome the shortcoming of poor quality of automatic answers [10]. The effectiveness of collaborative learning has been widely known. Currently, in the community-based learning approach, the community members are roughly divided into experts and novices. Knowledge flows in one direction from experts to novices through the process of Legitimate Peripheral Participation [3]. It is founded on an assumption that experts will always rush to assist novices in need of help. This is not always true in many cases because experts are often busy and they often need the volunteers to help the novices. Moreover, those experts are not absolute experts, which in fact do not exist. Knowledge from multiply domains evolves and technology is changing rapidly, the expert is at best a relative attribute of a specific context, which means a person is an expert in one context may become a novice in another [9].

Our community system is developed as a platform for users to ask and answer questions. Figure 3 shows QA flow in the community.

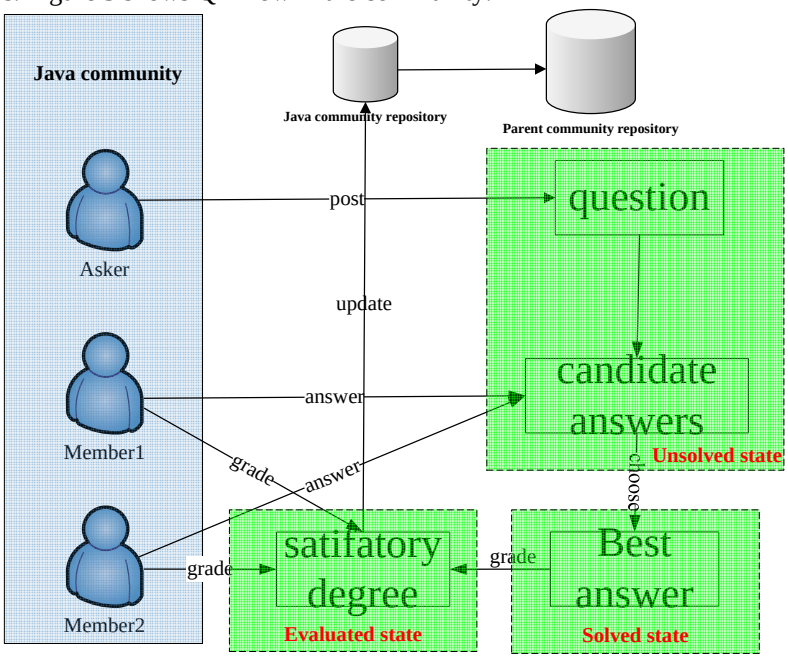

Fig. 3. User-interactive QA System Workflow

Step 1: In Java community, a member can ask some questions of which answers will be firstly searched in the repository. If there is not any suitable answer, the question together with the question type, rewarding score and deadline will be posted in the community.

Step 2: The community members can answer the question before the deadline.

Step 3: Some of answers can be selected as best answers by the asker or a ballot system, whose voters are the members in this community.

Step 4: The best answer has an important attribute named as the satisfactory degree, which represents the answer quality evaluated by community members.

Step 5: The pair of question and answer, regarded as a list of knowledge record, will be stored in the repository of the community.

From step 2 to step 4 is the lifecycle of a question. Unsolved state is from a question posting to candidate answers responses, which is followed by a solved state after choosing the best answer. After the processing of grade satisfactory degree among community members, it will turn into the evaluated state.

During the lifecycle of the question, we propose a user reputation mechanism. Such a mechanism can be an incentive rewarding for an individual to answer questions in the community. Because the answers are often judged by the member's user reputation, the answer quantity would be better than the traditional answer extraction mechanism.

For a single user, like Tom, he can participate in multiple communities and play different roles. According to his practices in different communities, he would be an expert in the programming community and a novice in the physics community. The community is defined as a deputy class of which instances are members. A single source object can have multiple deputy objects as long as it satisfies the selection conditions of communities. In this way, the same user can play different roles and have different reputations in different communities.

The reputation value is calculated according to three aspects:

1. Number of member's answers. If one member answers or asks more questions in a community, it means that he has more experience in this field. Thus, the member reputation is related to the number of questions and answers he posted in the community.

$$R_1 = \begin{cases} 1, & n > A \\ \frac{n}{A} & n <= A \end{cases}$$

n represents the one member's number of answers in the community. *A* represents a constant, often called punishment number. If *n* is less than *A*, the reputation would be weakened.

2. Member's academic background. Academic background is another important factor. As for the same answer, a core leader's grading weights more than the ordinary members. Thus we chose a symbol R_2 to represent this factor. R_2=B_1(core leader), B_2(active member),B_3(common member).Different academic backgrounds are assigned with different R_2 values, and R_2 is an decimal fraction between 0 and 1.

Those two factors above are taken into consideration from the point of member themselves, we define $W_1(u)$ to represent this kind of weight, Thus

$$W_1(u)=k_1R_1+k_2R_2,\ k_1+k_2=1.$$

3. Member's answer quality. If a member's answer is selected to be the best one, and its corresponding rewarding score is "s", if the value of "s" is bigger, it indicates that the user receives more approval in this question. Therefore, a member's answer score is also useful to portray the member's authority.

$$W_2(m)=\sum_{i=1}^{n}(mi)*Si\%$$

In this formula, n represents the number of one member's best answer in this community, m_i represents the question's rewarding score, and S_i represents satisfactory degree for a best answer given by the whole community, which is calculated as follows:

$$S_i=(\sum_{j=1}^{n}WjSj/n+S1)/2,$$

In this formula, *Wj* represents the user reputation of member *j*; *Sj* is the other member grading score to this answer; The process of grading satisfactory degree is mandatory for the leader and the asker in the community while optional to common members; *n+1* is the number of grading members in that community; *S1* represents the satisfactory degree given by the asker.

To sum up, user's reputation can represent $W= W_1(u)* W_2(m)$.In each community, there is a least-reputation, which is calculated according to the every member reputation in this community. To prevent rogue users from undermining answer process, members whose aggregate reputation score is blow least-reputation will be suspended from answering question in their communities, but still allowed to browse or ask questions in their communities.

5 Forestalling-answer System

We design a forestalling-answer system on the basis of Ajax chat room. It can provide both private and group discussion among members. More importantly, members can synchronously ask and answer questions in this text-based chatting room. A record of message consists of six attributes:

Message=<time, C_{id}, receiver name, sender name, message id, type>

The "time" is the message sending time, and "C_{id}" is used for identifying messages between different communities' contests. The attribute message type has a value of set of {SAY, QUESTION, ANSWER}, among which "SAY" indicates that message is a common message, "QUESTION" indicates that message is a question and "ANSWER" indicates that message is an answer to the question.

In the user-interactive QA system in the section 4, a record of knowledge can be defined as:

Knowledge=<question and answer pair, satisfactory degree, type>;

Knowledge consists of a pair of question and answer, a satisfactory degree to evaluate answer quantity and a type to represent the domain of the knowledge. If we take the community as a unit of analysis, rather than in a view of individual learning, there are various drawbacks in the community. First, those members except from the asker and responder would not acquire knowledge, because a pair of question and answer is usually generated from only two or three people interactions. Second, a sense of community togetherness, which would offer familiar comfort of a hometown

experienced by every member [7], would be gradually weakened if there are only few active member participations.

To solve the above problem, we propose a forestalling-answer system for members to attend. Figure 4 presents the workflow of the forestalling-answer contest.

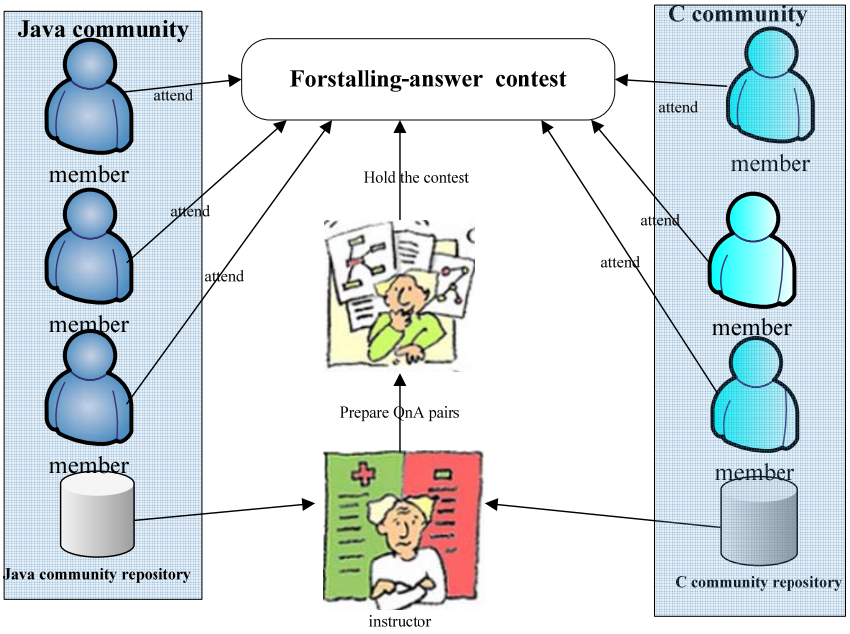

Fig. 4. Forestalling-answer Contests System Workflow

Step 1: Instructors prepare the question and answer pairs for the forestalling-answer repository. They would firstly rank the question and answer pair by satisfactory degree, and then those whose satisfactory degrees were low would be eliminated. Besides, those answers which are much related to an individual and out of the community domain knowledge would also be eliminated by the instructor manual filter.

Step 2: After the question and answer pairs have been prepared, the instructors would hold a challenging and well-planned forestalling-answer contest. All of the community members would be invited to be the opponents. Instructors would make a plan of the rewarding score and fix an appropriate time when all the members are available.

Step 3: During the contest, the instructor asks the questions and sends them to online opponents. The opponent would win the rewarding score if he or she can firstly give the right answer.

Step 4: If the instructor could be the organizer of the parent community, for example the programming community which consists of two sub communities. Since those two communities have common ground, such as algorithm. He could also prepare the question and answer pairs which consist of half from c community and the

other half from Java community, and then invite all the members in these two communities and finally organize a forestalling contest between them.

Step 5: In the contest of two communities, opponents are divided into two contest groups according to their belonging communities. The winner group depends on the number and rewarding score of right answer provided by community members.

By participating in foresting-answer contest, most members would acquire knowledge, which would advance acquisition of community knowledge as a whole. On one hand, those who give the right answers would review their knowledge. On the other hand, by immersing themselves into the processing of considering the answers to the questions and then comparing them with the right ones, those active thinkers would also learn knowledge even if they fail to give the right answers. By holding a contest between communities, members would experience a sense of togetherness. Opponents from each group would actively contribute their ideas and cooperate with other members to deal with the questions. Their practices of answering would be active, which would thrive the community collaborative learning as a whole.

6. Conclusion

In this paper, we proposed a flexible community based on Object Deputy Model [8]. It is easy for bigger community to be divided and small communities to be merged. We provide a community recommendation mechanism for user to easily find their communities. To enhance the collaborative learning in the community, we provide a user-interactive QA system together with a forestalling-answer contest system. In the user-interactive QA system, we design a user reputation model for incentive participation and a formula to evaluate answer quality. In the forestalling- answer contest system, the sense of togetherness can be inspired and the whole community ability to knowledge acquisition would be improved.

7. Acknowledgments:

This research is supported by National Natural Science Foundation of China (60573095) ， the Program for New Century Excellent Talents at University of China (NCET-04-0675)， Hi-Tech Research and Development Program of China (2006AA12Z210), Research Fund for the Doctoral Program of Higher Education(20050486024), the Key Project of Chinese Ministry of Education (107072) and National Basic Research Program of China (2007CB310806).

8. References

1. Stahl, G., Koschmann, T., & Suthers, D.: Computer-supported collaborative learning: An historical perspective. In R. K. Sawyer (Ed.), Cambridge handbook of the learning sciences. (pp. 409-426). Cambridge, UK: Cambridge University Press (2006)

2. Chen, W.: "Supporting Teachers Intervention in Collaborative Knowledge Building". Proceedings of 16th European Conference on Artificial Intelligence (ECAI'2004). Workshop on Artificial Intelligence in Computer Supported Collaborative Learning. Valencia, Spain. E.Gaudioso & L.Talavera (eds). pp. 1-5 (2004)
3. Lave, J. & Wenger E.: Situated Learning: Legitimate Peripheral Participation Cambridge: Cambridge University Press. (1991)
4. Lave, J.: Situating learning in communities of practice. In L. Resnick, J. Levine & S. Teasley (Eds.), Perspectives on socially shared cognition (pp. 63-83). Washington, DC: APA (1991)
5. Stahl, G.: Group cognition:Computer support for building collaborative knowledge. Cambridge, MA: MIT Press. (2006)
6. Sfard, A.: On two metaphors for learning and the dangers of choosing just one. Educational Researcher, 27 (2), 4-13. (1998)
7. Etienne Wenger, Richard Arnold McDermott, William Snyder: Seven Principles for Cultivating Communities of Practice. In Etienne Wenger, Richard Arnold McDermott, William Snyder, Cultivating Community of Practice:A Guide to Managing Knowledge. Harvard Business School Press. (2002)
8. Zhiyong Peng: An Object Model for Advanced Database Applications, PHD Dissertation in Kyoto University. (1995)
9. Yunwen Ye, Yasuhiro Yamamoto, Kouichi Kishida: Dynamic Community: A New Conceptual Framework for Supporting Knowledge Collaboration in Software Development," apsec, pp. 472-481, 11th Asia-Pacific Software Engineering Conference (APSEC'04). (2004)
10. Wanpeng Song, Liu Wenyin, Naijie Gu, Tianyong Hao: Automatic Categorization of Questions in a User-interactive Question Answering System. WWW2008 workshop QAWeb2008. (2008)
11. Xiaojun Cheng: Research on Question and Answer Technology in Web Community, Master Dissertation in Wuhan University. (2008)