A COMMUNITY OF PRACTICE APPROACH TO LEARNING PROGRAMMING

Gwo-Dong Chen, Liang-Yi Li, and Chin-Yea Wang
No.300, Jhongda Rd., Jhongli City, Taoyuan County 32001, Taiwan(R.O.C.), Department of Computer Science and Information Engineering, National Central University
Corresponding author: Liang-Yi Li
No.300, Jhongda Rd., Jhongli City, Taoyuan County 32001, Taiwan(R.O.C.)
E-Mail Address: lihenry12345@gmail.com

ABSTRACT
In programming courses, teaching students who have varied levels of knowledge and skills the requisite competencies to perform in real-world software development teams is indeed difficult. To address this problem, this paper proposes a community of practice (CoP) approach and provides some guidelines to simulate a real-world CoP in a blended learning environment. It simulates not only the role structure and tasks but also the participation process. The design of this approach is based on the situated learning perspective that considers learning to be a trajectory in which learners move from legitimate peripheral participation to full participation in CoPs. The results from the data analysis and questionnaires indicated that the students were very engaged in this activity and believed that this approach helped them to develop necessary programming knowledge and skills. This study also reports certain constraints and suggestions for teachers who want to adapt this approach for their courses.

INTRODUCTION

One of the most important skills for the computer science student is programming. In general, computer science students first learn programming skills in an introductory computer science course, often termed CS1. The students who enroll in this course possess widely different learning attitudes and programming backgrounds, therefore making it difficult for the instructor to provide a learning environment that benefits all students. Moreover, learning to program involves acquiring programming knowledge and developing various skills that are required in real-world software development processes such as designing, planning, coding, and testing (Robins, Rountree, & Rountree, 2003). First-year computer science students face many challenges in learning these valuable soft skills and acquiring the requisite knowledge. Traditional teacher-centered instructional approaches aimed at averaging the majority of students are not effective. To be effective and to benefit all students, teachers usually must employ other learning activities in their programming courses to complement their instruction.

To support learning in the programming courses, this paper proposes a community of practice (CoP) approach. This approach provides guidelines to simulate a real-world CoP in a blended learning environment. It simulates not only the role structure and tasks but also the participation process. In the simulated community, each student is assigned a role based on his/her learning status. Each role has its respective tasks and responsibilities and incorporates different skills. When the individual who is assigned that role has demonstrated competency in certain predefined criteria, he/she will be promoted to more central roles. In this way, all students have the opportunity to assume every role and to develop the different levels of knowledge and skills associated with each of the independent roles. Because this approach requires out-of-class effort and work, a Wiki system is used to design an online learning environment that provides students an interactive, knowledge-sharing space. We have implemented this approach in a programming course. In this paper, we report our findings and suggestions.

1. RELATED WORK

1.1. Situated Learning

In recent decades, the focus on education has changed from an acquisition metaphor to a participation metaphor. The acquisition metaphor is based on cognitive theories and assumes that knowledge consists of symbolic mental representations in which learning is a process of acquiring and manipulating these symbols (Sasha A. Barab & Duffy, 2000). In contrast to learning as knowledge acquisition, the participation perspective is grounded upon situated learning (Lave & Wenger, 1991); that is, it views knowledge as situated in the activity, context, and culture in which it is developed and used. As such, learning becomes a trajectory in which learners' transition from legitimate peripheral participation (LPP) to full participation in CoPs. Learning occurs in a participation framework rather than in an individual mind.

CoP and LPP are two major characteristics of situated learning. A CoP is a living context in which learners with a common interest in a subject collaboratively share ideas, find solutions, and solve problems. It serves as both context and content, providing support for learners as they acquire appropriate skills, knowledge, experiences, beliefs, and values related to the CoP (S. A. Barab, Barnett, & Squire, 2002; Henri & Pudelko, 2003; Wenger,

McDermott, & Snyder, 2002).

Learning in a CoP is attained by increasing the access of learners 'to participating roles in expert performance' (Lave and Wenger, 1991, p. 17). When newcomers enter a CoP, LPP provides them with the opportunity to imitate and observe old-timers and motivates them to engage in community activities. In this way, the newcomers are able to form a general idea of what constitutes the CoP. Gradually, they are given more responsibilities and more complex tasks within the community, and they begin to engage in practices that are more central to the operations of the CoP. As they do so, they monitor and adjust their motivations, knowledge, skills, identities, beliefs, and values. They also have increased opportunities to interact with community members on multiple levels. The interactions on these various levels affords the member multiple and diverse opportunities for learning and contributing to the community (Wenger, 1998). Eventually, as they become more experienced in the CoP, they develop increased skill levels and greater knowledge and reach old-timer status. Fig. 1 shows how a newcomer moves from peripheral to central participation as well as how he/she changes his/her roles and tasks within the community.

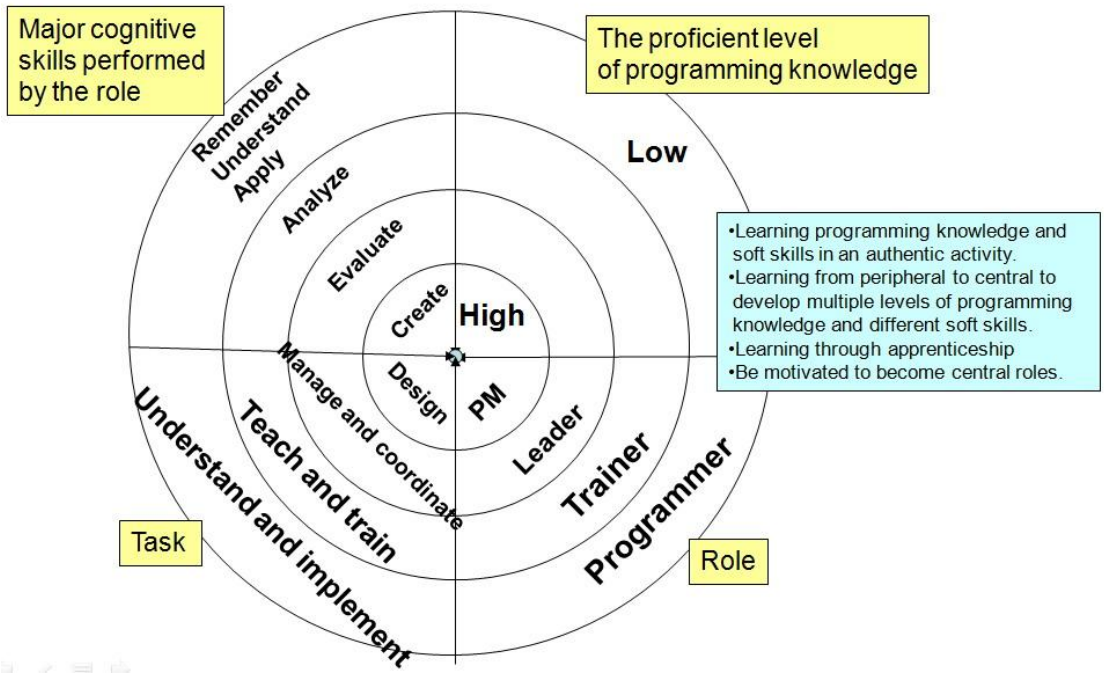

Fig. 1. Learning Transitions from Peripheral to Central

According to the above descriptions, we summarize advantages that situated learning provides the learner. These advantages, listed below, are inherent to most CoPs.

1. Learning practical knowledge and skills in an authentic context.
2. Learning transitions from peripheral to central participation, developing multiple levels of knowledge and different soft skills.
3. Learning through apprenticeship.
4. Being motivated to assume central roles through increased participation.

1.2. Learning activities designed for supporting programming courses

Several approaches have been proposed for improving CS1 courses. For example, the apprenticeship-approach encourages students to read and extend programs written by experienced programmers (Astrachan & Reed, 1995; Kölling & Barnes, 2004). The peer-assessment activity allows students to read, review, and assess each other's programs. (Hamer, Purchase, Denny, & Luxton-Reilly, 2009; Zeller, 2000). The group cooperation approach, which divides students into groups to solve real-world problems or to complete a program, is also widely used (Hsieha & Jr., 2002; Joy, 2005; Kalles, 2008).

Each of these approaches has value. However, none of them takes into account the fact that students' abilities and attitudes towards learning are varied. Students with a range of approaches to learning are all assigned the same tasks or roles, though some students may not be able to complete the assigned tasks or to play the assigned roles. Furthermore, the more passive students may plagiarize from peers or become free-riders in the cooperative activities (Raban & Litchfield, 2007; Strijbos, Martens, & Jochems, 2004).

To address these concerns, information and computer technology (ICT) systems have been incorporated to develop adaptive learning systems and to construct collaborative learning environments in which students support each other and share knowledge to achieve common goals. These systems and environments are designed based on the work of Vygotsky, who proposed the zone of proximal development (ZPD) (Vygotsky, 1978). Li & Chen (2009), for example, developed an adaptive coursework support system to provide students adaptive programming exercises and learning support. Chang, Chen, & Li (2008) designed a web-based coursework environment for students to share and review programming knowledge in an online community.

Wiki-like systems such as MediaWiki (http://www.mediawiki.org/wiki/MediaWiki) and CoWeb (Guzdial, Rick, & Kehoe, 2001) are generally used as Web-collaborative platforms. They allow any user to edit any existing page or to create new pages. They make it easy for teachers and students to create collaborative Web-based activities. Some studies have used Wiki systems to support programming courses. For example, Xu (2007), in a compiler construction course, used a Wiki system to support team project management, and Bennett (2009) used a Wiki system as the medium for a student-authored CS1 programming textbook. These studies indicate that using Wiki in programming courses can generate high quality content, improve knowledge sharing, and support communication and coordination tasks. However, other studies have found that only providing Wiki systems did not facilitate collaboration, as students using the Wiki systems still largely worked as individuals. Therefore, they further suggested that Wiki systems must be well-integrated with collaborative activities, curriculum, and assessment (Choy & Ng, 2007; Cole, 2009; Judd, Kennedy, & Cropper, 2010; Karasavvidis, 2010; Weaver, Viper, Latter, & McIntosh, 2010).

Because of the advantages of situated learning, this paper proposes a CoP approach to simulate a real-world CoP in the classroom. In the simulated community, students with varied abilities will support each other to achieve common goals and will be provided opportunities to learn multiple levels of programming knowledge and skills. In addition, a Wiki system is used to support this approach outside of the normal classroom hours. We believe that integrating the Wiki system will motivate students to participate in collaborative activities and will further help them to develop course knowledge and soft skills outside of class.

2. GUIDELINES FOR IMPLEMENTING THE COP APPROACH

A CoP is an open environment that allows anyone to join and leave at any time. Member identity is formed naturally based on one's level of involvement. However, a learning community is, typically, a closed community. All students are newcomers at the beginning of a course, and one cannot join the community without the teacher's approval. The identity of students cannot be formed naturally. Therefore, it is difficult to simulate the participation process of a real-world CoP in the classroom. We provide several guidelines for teachers to create an activity that simulates a real-world CoP with the legitimate peripheral participation process. These guidelines are as follows:

1. The structure of the roles must be designed to be hierarchical, and those who play more central roles must assume those tasks that involve higher level knowledge and skills.

Because the role structure is hierarchical, students can be endowed with different roles based on their learning status. Greater higher-level knowledge and skills are required for the central roles, and greater responsibilities are associated with these more central roles. Therefore, those students who perform well and display strong skills should be promoted and assigned to the more central roles. The central roles should be filled by the more capable students, as they guide those in the peripheral roles to collaboratively complete common tasks. Those in the peripheral roles should observe and imitate those in the central roles, thereby developing and enhancing their knowledge and skills.

2. In the beginning of a course, it follows that the teachers and teachers' assistants will fulfill the more central roles.

At the beginning of the course, all students are, essentially, beginners. The only experts are the teachers and the teaching assistants (TAs). Therefore, teachers and TAs must play the central roles at the beginning of a course, thus placing greater demand on the workload of teachers and TAs. To reduce the demand on the teachers, teachers often enlist the aid of capable students who successfully completed the course the last year. Teachers also can conduct activities to assess the skills and knowledge of students and then enlist the more capable students in the class to assume the responsibilities associated with more central roles.

3. A role promotion mechanism that can assess students' skills and determine students' roles should be designed.

In a real CoP, member roles are formed naturally based on levels of involvement. Because a learning community is closed, however, all students are newcomers at the beginning of a course. To determine the role of each student, a role promotion mechanism should be implemented. Accordingly, students can be promoted to more central roles if and when they demonstrate or master predefined criteria.

4. This approach should be implemented within the context of the class on a regular basis, similar to that of regular school work and weekly coursework.

- Students' roles must be adjusted over time, and the teacher must have a preconceived idea of what constitutes suitable or adequate time. It is our conclusion that a student's skills and knowledge should be assessed at the end of an old assignment and at the beginning of a new assignment. This allows the teacher to reference the students' previous learning performance based on the old assignment for role promotion, and students can use new roles to complete the new assignment. If this is the case, we must separate this activity into several assignments. After each assignment, students should be given the opportunity to change their roles. Therefore, we suggest that this approach should be implemented in the context of regularly school work, such as weekly coursework. Through the assessment of the students' regular school work, the teachers can routinely check students' statuses and adjust their roles.

5. Teachers must help students to understand and recognize the tasks, responsibilities, and objectives of each role.

When students sufficiently understand and recognize their roles and tasks, they will focus on their responsibilities and their contributions to the community as a whole. Therefore, teachers must help students to understand and recognize the responsibilities and tasks associated with each role, the objectives of each task, and the advantages they can gain when playing a specific role (e.g., what rewards they can earn and what they can learn). Various strategies can be used toward this end. For example, teachers can introduce the roles at the beginning of the course and publish the evaluation rubric for each role on the Internet. They also can publicly praise the students who have performed a role well or correct the mistakes of the students who have failed to perform a role successfully.

6. Teachers must encourage students to engage in the activity.

Not all students in a class are willing to engage in a learning activity. Teachers, therefore, must encourage all students to engage in the learning activity, and some strategies for this exist. For instance, teachers can structure positive social interdependent communities (Johnson, 2003; Johnson & Johnson, 2002). In addition, detailed information on the performance of all students in the activity can be accessible on a Web page, thus subjecting students to peer pressure and/or encouragement. Furthermore, teachers can send an alert by email to notify students regarding the status of various tasks, for example, which task is overdue and which task must be finished.

7. ICT can be used to provide an authentic and interactive context for outside classwork.

In general, students attend a course only two or three hours a week. This is not enough time for teachers to conduct such an in-depth collaborative activity. Therefore, the approach must be implemented outside of the normal classroom environment. ICT can provide interactive Internet and multimedia applications to simulate realistic situations and to connect with remote persons and applications. It also can be used to create authentic contexts and to support students' interactions and discussions with peers outside the classroom walls.

3. CASE STUDY

We have implemented this approach in a mandatory course entitled 'Basic Computer Concepts' in which fifty first-year computer science students studied the C# programming language. One teacher lectured for three hours each week in the classroom, and one teaching assistant facilitated the students' participation in an out-of-class activity based on the above-mentioned approach in the Web environment.

3.1. The Activity

The learning activity simulated the role structure and process of participating on a software development team. A software development team is typically composed of a project manager and one to three small teams; each small

team is composed of a team leader, a trainer, and two or three programmers. A project manager must design a group assignment for his/her small teams. Each small team then conducts a software development process, planning, design, coding, and testing to complete a program for each assignment. There are specified tasks for every role. When working on these tasks, members are given responsibilities and are required to perform the corresponding skills. For example, a team leader must distinguish the parts of a group assignment, divide it into several subtasks for his/her members, and organize each member's task outcomes into a final report. Thus, the team leader must use the cognitive skills of analysis and synthesis to accomplish his/her tasks. Figure 1 represents the roles, tasks, and learning process in the activity.

The activity was used for coursework, and assignments were issued every two weeks. The software development process was used for every assignment. Every two weeks, the teacher determined the topic of an assignment according to the course syllabus. Before the students began the assignment, each project manager was required to design a group assignment for his/her small teams. The remainder of the two-week period was used for the assignment (during the first week) and for reviewing and revising (during the second week). Before the end of the first week, the team leader was required to integrate all members' task outcomes into a final report and submit it via a Wiki system for later review.

The agenda for the second week was scheduled to include three days for an expert assessment and four days for revisions. After the submission deadline, each project manager reviewed the submitted programs for his/her small teams and provided comments about the end result after which each small team revised its program according to its project manager's comments. Finally, each small team submitted the revised program via the Wiki system and demonstrated the program for the class.

After the assignment, the teacher adjusted the students' roles for the next assignment based on the role promotion criteria, including students' performance on the last assignment, the results of a peer assessment that was conducted at the end of each assignment, and the teacher's observations.

In addition to working on group assignments, students were also encouraged to share their knowledge and learning on the Wiki system, such as what they had learned, problems they had encountered, and websites they had found useful. The individual contributions were also rewarded.

3.2. The Procedure

Fourteen assignments were given over two semesters. Students were allocated two or three weeks for each assignment. Seven assignments (1~7) were given in the first semester, and seven assignments (8~14) were given in second semester.

At the beginning of this course, student learning status was unknown. Therefore, the teacher used three assignments to probe the learning status of each student. For each of the first through third assignments, the teacher prepared three alternative programming exercises for all students. After these three assignments, the teacher divided all students into ten small teams, according to their learning statuses. Ten students were promoted to the role of 'team leader,' and ten students were identified as 'trainers'. At that moment, the teacher and the teacher's assistant played the role of 'project manager'. For the fourth through eighth assignments, the teacher assigned a group assignment to each team. Each small team was required to complete a program by a given deadline. After these five assignments, the teacher selected the individual most capable of playing the role of 'project manager' from the team leaders. Thus, the software development teams were formed. Later, all of the roles were played by the students. The teacher and teacher's assistant only monitored the flow of the activity.

3.3. *The Web learning environment*

The implementation of the Web environment used MediaWiki. It provided support for teachers to manage the flow of the activity. It also provided support for students to manage assignments and personal and group information, to discuss with peers, to share knowledge, and to inquire about their status on the Web. In addition to the basic pages including history and discussion pages in each Wiki, there are four kinds of pages the students and TA create for the activity.

- The course announcement page: This page is created by the TA. It is the homepage of the Wiki system, and it is used for class announcements such as schedules, timelines,, teaching materials.
- The assignment page: The procedure of a group assignment is presented in a series of steps: designing a group assignment, planning a schedule for the assignment, submitting the completed assignment, and evaluating the submitted assignment. All information about the assignment, such as the assignment description, task arrangement, task outcomes, integrated final report, and comments on the submitted

assignment is published on the assignment pages. Fig. 2 is a snapshot presenting an assignment page.

- The group and personal page: Each team has its own group page in which members can post their group portrait. There are four types of information that must be included on a group page: a group picture, a basic introduction of each member (name, role, and programming experience), hyperlinks to each member' s personal page in which a student can post his/her portrait, and hyperlinks to each assignment page of the small team. The group and personal pages display the group and personal portraits and other information about the group to promote positive identity interdependence.
- The knowledge sharing page: Both instructor and students can freely create and modify pages in this section. These pages serve as common accessible source for sharing and exchanging knowledge. Students are encouraged to build and share knowledge on this page.

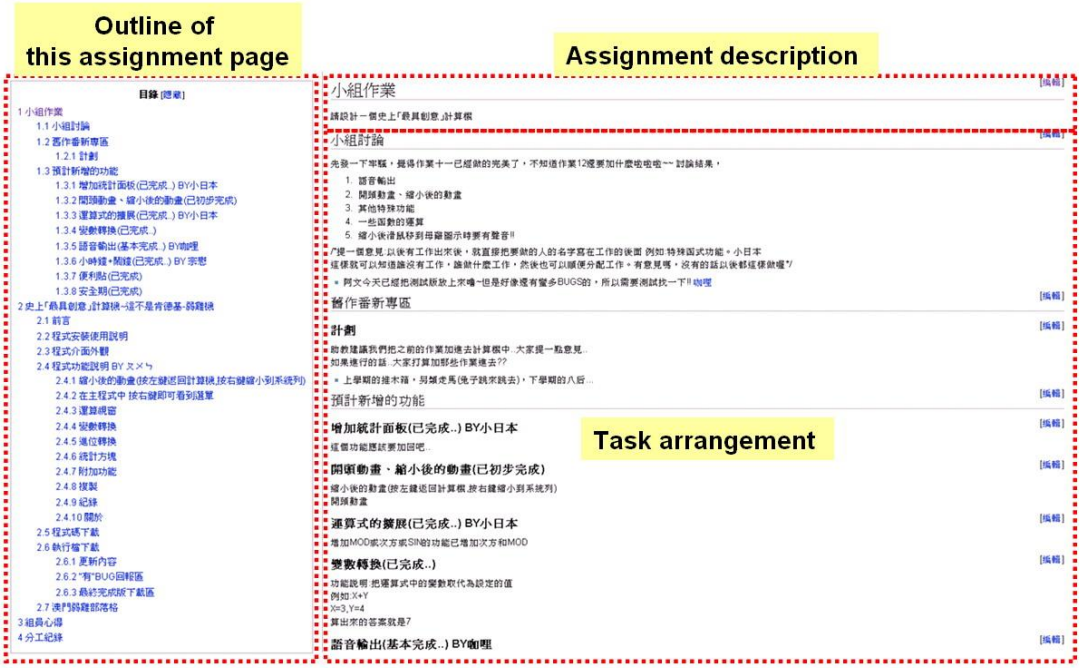

Fig. 2. A snapshot of a posted group assignment

3.4. Data Collection

To understand the effect of this approach on student learning and collaboration, we collected a combination of quantitative and qualitative data that included the following: (1) students' portfolios, which the students posted on the Wiki system, including discussions with other students, completed assignments, and students' comments related to each assignment; (2) a structured questionnaire that the students completed at the end of the course, based on a Likert scale rating system (1=Completely Disagree, 5=Completely Agree); and (3) notes from semi-structured interviews and informal discussions with students. The data were analyzed to answer the following questions.

1. To what degree did the students participate in this activity?
2. Did the students perceive that this approach helped them develop soft skills?
3. What were the roles of the TA?

4. RESULTS

4.1. Participation in the activity

From interviews, we found the students usually met two times in group face-to-face discussions for each assignment - once for the division of labor and once for the assignment integration. There were two ways of dividing the labor that the students performed. 1) The leader divided the assignment into several subtasks. Then the leader and trainer individually guided one or two programmers to complete the subtasks. Finally, the leader and trainer worked together to integrate the outcomes of the subtasks into the final report. 2) The leader first divided the assignment into several subtasks. Then he/she monitored the progress of each programmer, while the trainer supported programmers to complete their tasks. Finally, the leader integrated the outcomes of the subtasks into a final report. The major consideration or difference between the two methods was related to the role of trainer and was based on the programming ability of the trainer. If the programming ability of the trainer was similar to that of the leader, the group adopted the first method. If the programming ability of the trainer

appeared to be lacking, the group employed the second method.

We also found that the diversity of the programming ability of members within the group affected how well and to what degree the students collaborated. If the programming ability of the members in a group was more equal, the positive interdependence of this group was high. However, if the diversity of the members was great, the members with lower programming ability relied heavily on other members of the team (usually the leader and trainer). As a result, these students did not do anything as members of the collaborative team and, thus, were unable to learn the requisite skills.

Our approach, which promotes capable leaders as project managers, can solve the problem. When a leader is promoted to the role of project manager, the remaining members have no one to rely on. Thus, it forces the members to complete the assignment by themselves. For example, one student commented on the Wiki after his leader was promoted to project manager:

> *I feel the workload of every member in this team has increased after Jimmy became the project manager. But I think it is good for me because I had to do more and had more opportunities to learn from doing my group assignments. Like this assignment, I was responsible for writing program code. I am very glad for completing the program.*

Additionally, the students also developed their own strategies to solve the problem. For example, after finding that some members did not do anything, the leader of the sixth group asked the members to complete the assignment by themselves. He just answered their questions and gave them suggestions. He announced the plan on the Wiki system

> *I will not do programming tasks in this assignment. I hope you can finish this assignment by yourselves. If you have any questions, I will answer them.*

We also found that there was a team in which none of the programmers were capable of completing their tasks in any of the assignments. Therefore, the leader actively asked the programmers to participate in an after-class meeting in which he taught the programmers how to program, thus, enhancing their programming knowledge and skills.

In addition to meeting face-to-face, the students also discussed and shared information on the Wiki system. The students posted and responded to the generated ideas, the task arrangements (division of labor), the task outcomes (the code and executed results of the program), the knowledge and skills that they learned, and the related resources used for the assignment in the assignment pages. We found that the students did not use the Wiki discussion page but, rather, posted questions directly on their group assignment pages for collaboration. However, they did not post any questions or comments on the assignment pages of the other groups. Most of the questions they discussed were related to the division of labor.

Table 2 presents descriptive statistics automatically generated by the Wiki system. In this study, the students created many pages for their assignments. The average number of edits per page is 5.5, which may indicate that the students often used the Wiki to record and discuss their assignment. In addition, the average number of pages read per user is 1525. This may indicate that the students thought the articles were valuable and were willing to read to enhance their learning. The descriptive statistics may also demonstrate that the students were engaged in the activity.

Table 1: The descriptive statistics in the Wiki system

Items	Number
The number of content pages that is authorized as high quality by MediaWiki system	216
The average number of edits per page	5.50
The average number of times each user read pages	1525.20
The average number of times each user edited pages	241.90

4.2. Role migration

Fig. 3 presents the numbers for each role in each assignment. Those in the peripheral roles migrate gradually to the central roles. At the end of the course, thirteen students had played the role of ‘project manager’, seven students had played the role of ‘team leader’, and eight students had played the role of ‘trainer’. In other words,

these students were recognized for their abilities, showing that they had been trained in their respective soft skills. It also represents that the promotion mechanism caused students to appreciate and understand their positions within the community. Students were then motivated and wanted to take on central roles.

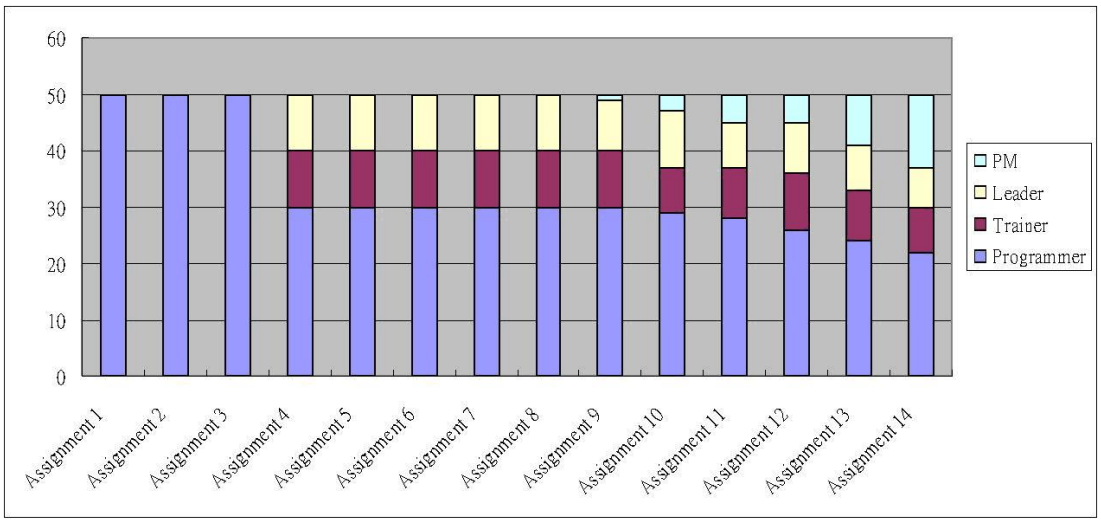

Fig. 3. Role migration

In addition to upgrading students' roles, however, six students were downgraded at some point during the process. There were three primary reasons that the six students were downgraded.

1. The leaders or trainers had other personal tasks such as after-class activities and/or part-time jobs and did not have enough time to take care of their responsibilities. (n = 2).
2. The leaders or trainers were incompetent. Their members suggested downgrading them. (n = 2).
3. The leaders or trainers thought that they did not have abilities to lead their members (n = 2).

When we first (assignment 4) divided the students into ten small teams, each team was composed of five students. However, at the end of the course, there were thirteen students playing the role of 'project manager'. The number of members in each small team was decreased (seven small teams consisted of four students, and three teams consisted of three students). Thus, students' workloads were increased on each team. Although the workload of every student was increased, we found that the students did not complain. In fact, they felt it provided them more opportunities to practice their programming skills.

Project managers were required to design and evaluate the group assignment. Thus, their workload was considerably less than other members of the teams. We found that in addition to answering questions, however, most of the project managers created a knowledge-sharing page where they gathered information and wrote articles related to C# programming in the pages. Some students responded that "*these articles are useful and I frequently read them for learning programming and completing my tasks.*"

4.3. Students' perception

A structure questionnaire that the students completed at the end of the course, using a Likert-scale rating system (1=Completely Disagree to 5=Completely Agree) was designed to investigate whether the students perceived that playing particular roles helped them to cultivate particular soft skills. The results are listed in Table 1 and indicate that the students highly agreed that they actually developed the skills by playing these roles.

From the students' portfolios, we found that there may be two reasons why the students developed their skills in the activity. First, the students learned from observing and imitating those in the central roles. For example, the team leader of the tenth team guided the team successfully to complete each assignment; thus, he was upgraded to project manager for assignment 11. Meanwhile, the trainer of this team was upgraded to team leader. We found that this new team leader divided and planned a group assignment in a way very similar to the previous team leader. The new leader told us that "*because the previous leader played the role very well, I imitated what he did*."

Second, the students can learn skills through interactions and collaborations with those in different roles. For example, a student was upgraded as a team leader even though he was not familiar with the tasks of the role in

the beginning. When collaborating with other members (the project manager, trainer, and programmers), however, he received comments and feedback from them and gradually began to understand what he should do and how he should do it. This student posted the following statement when he first assumed the role of "team leader":

> *I want to thank my project manager, as he helped me very much in this assignment. But I think I am not suitable to play the role of "team leader" because I cannot guide my members to complete a good program.*

After he played the role of team leader two times, he posted the following statement:

> *I very much appreciate the TA who gave me the opportunity to become a team leader so I can learn how to guide a team to successfully complete a program and how to coordinate with my team members. I also appreciate my members. They always gave me some significant comments to let me know what I should do and how I should do it, although there were some conflicts between us.*

Table 2: Students perceived that playing these roles is helpful for developing skills

Question (the questionnaire has been translated from Chinese)	Mean	SD
Answered by those students who have ever played the role of 'Project Manager' (N=13)		
Reviewing peers' assignments is helpful in cultivating my evaluation ability	3.50	0.80
Designing group assignments is helpful in cultivating my creativity	3.83	1.03
Answered by those students who have ever played the role of 'Team Leader' (N=21)		
I can learn how to divide a group assignment based on every team member's ability when I am a team leader	4.33	0.66
Answered by those students who have ever played the role of 'Trainer' (N=20)		
I can learn how to guide members in completing their tasks when I am a trainer	3.55	0.69
Answered by those students who have ever played the roles of PM, Team Leader, or Trainer (N=32)		
I can learn various skills (e.g., designing, planning, evaluating, and coding) by playing different roles	4.09	0.80
Answered by all students (N=50)		
Comparing with self-study, I learned more in this activity	4.12	0.80
Participating in the activity is helpful for learning programming skills and knowledge	4.20	0.57
Participating in the activity is helpful for learning communication and social skills	4.12	0.63

4.4. Teacher assistant's roles

The students encountered many learning and emotional obstacles while participating in the activity. The TA needed to provide a rich context complete with the needed resources and also was required to solve students' problems. Without the support of the TA, the activity could not be successful. We summarize four kinds of roles that the TA played in this activity. First, the TA is the old-timer, the one with previous experience and knowledge. At the beginning of this course, all students were newcomers; the only old-timer was the TA. Therefore, she was required to assume the role of 'project manager', 'team leader', and 'trainer' and transfer her experiences to the students. Second, the TA is the mediator of conflicts. Members of a team bring a diversity of experiences and perspectives, thus creating a learning environment in which task-related and relationship-related conflicts often occur. The TA must continuously monitor each team and resolve their conflicts efficiently and effectively. Third, the TA must provide students with emotional support. The students' roles may change from one assignment to the next. Some emotional challenges may arise as a result of these changes. The TA must resolve these problems and provide the necessary support. For example, when one student is downgraded, he/she may be frustrated. The TA must pacify this student and let him/her know why he/she was downgraded and then encourage him/her to work harder on the next assignment. Fourth, the TA is the key person to facilitate students' participation. She not only encouraged interactivity and contributions but also invited students to guide the more silent, passive students to join in the activity. For example, the TA promoted some students to assume the roles of leaders and trainers as they were viewed to be capable and enthusiastic students. These students then guided and encouraged the programmers (silent students) to participate in the activity.

5. DISCUSSION AND CONCLUSIONS

Situated learning depicts learning as a participation process in which learners move from legitimate peripheral participation to full participation. In this process, learners gradually become more involved in the community activities and, thus, their motivations, values, and identities change. The legitimate periphery provides learners the appropriate context to understand the community. The identity change gives them new roles, tasks and responsibilities and provides them the opportunity to develop new knowledge and skills required by the

community. Without the legitimate peripheral participation and identity changes, learners do not have the opportunity to become the experts.

Based on situated learning, this study proposed a CoP approach that provides some guidelines for simulating the role structure and participatory process of a software development team in the Wiki system. The teacher endowed each student with a role according to his/her learning status. Based on these roles, students took responsibility for their tasks and mastered new skills by observing and imitating more experienced, more knowledgeable members and by collaborating and interacting with those in other roles. The results indicated that the students were very engaged in this activity and perceived that this approach provided a context that helped them to develop soft skills and programming knowledge. Although the approach was implemented in a programming course in this study, we believe that it can also be used in other courses in which students need to cultivate various real-world skills.

While this approach provides some advantages, it also demonstrates several constraints. First, some project managers did not think that reviewing peers' assignments enhanced their evaluation ability (mean=3.50); some trainers did not think that they could learn how to guide members to complete their tasks (mean=3.55). The role of the project manager is most central. Unlike those in the other three roles, the project managers received less support and less feedback from other students. Therefore, they usually could not understand whether their evaluations and comments were acceptable when they evaluated their peers' assignments. In addition, teaching or helping peers to solve a problem is a complex process. It is necessary that instructors continuously monitor their learners' levels of self-regulation and then offer proper scaffolding to assist students through the zone of proximity development (Tharp & Gallimore, 1988). To solve these problems, we suggest that teachers provide clear guidelines for every role such as how to review peers' assignments, how to use different approaches to help members complete their tasks (Johnson & Johnson, 2002), how to offer adapted feedback to let students know how well they have performed in a task and how to improve for different roles, especially for that of the project manager (Hsieha & Jr., 2002).

Two issues must be considered when promoting and changing roles. One is that the workload of the peripheral roles may have increased significantly after more students have taken on central roles. Although the students did not complain about this, we suggest that teachers or project managers consider students' workloads when they design a group assignment. Sometimes, teachers can also ask project managers to support these teams that were incapable of completing their assignments. The other issue that requires attention is the effect of downgrading a student as he/she might become frustrated. To reduce students' negative reactions, we suggest that the teacher privately talk to the student and explain that he/she will be downgraded on the next assignment and then encourage him/her to work hard on the next assignment. Furthermore, it is advisable to acquire his/her permission before posting the roles for the next assignment on the Wiki system.

Some students felt role conflict. A classroom community is one kind of CoP. A student participating in a classroom community experiences a process change from peripheral to central participation. In the beginning, all students are recognized as the same identity "newcomer". After participating for a period of time, students will be tacitly endowed with different identities according to their learning performance, personality, and social skills. . In contrast to the classroom community, our approach explicitly endows roles for each student. Some students are endowed as project managers, thus they are recognized as central participants. Some students are endowed as programmers, thus they are recognized as peripheral participants. The tacit roles in the classroom community and explicit roles in our activity make some students exhibit or experience role conflict. For example, some members thought that their leaders or trainers did not have the abilities to effectively lead them, and some leaders or trainers though that they did not have the requisite abilities to lead their members.

The TA is the most important person for the success of the activity. He or she not only assumes the role of the old-timer in the beginning of the activity, but also acts as the intellectual, social, managerial, and technical facilitator throughout the whole process. Assuming these roles greatly increases the workload. On average, the TA spends two hours per day reading students' articles, providing comments, answering their questions on the Wiki system and discussing with students via instant messaging or face-to-face. However, the TA's responsibilities gradually decrease as more students begin to fulfill central roles.

The students primarily used the Wiki to present the outcomes of their assignments. The number of discussions in the Wiki system remained nominal. Most of the questions posted were related to the division of labor or how to use and navigate the Wiki system. Very little discussion was related to programming knowledge. The infrequent postings may be caused by the nature of the Wiki as it appears more suitable for collaborative writing tasks than

for discussion (Choy & Ng, 2007; Elgort, Smith, & Toland, 2008). To support asynchronous discussion, the threaded discussion forum remains the best option.

ACKNOWLEDGEMENT

This project was supported by the National Science Council of Taiwan under contract numbers NSC 100-2811-S-008-001 and NSC 99-2631-S-008-004.

REFERENCES

Astrachan, O., & Reed, D. (1995). AAA and CS 1: The Applied Apprenticeship Approach to CS 1. In C. Laxer, C. M. White, J. E. Miller & J. Gersting (Eds.), *Proceedings of the twenty-sixth SIGCSE technical symposium on Computer science education*. New York: ACM Press, 1-5.

Barab, S. A., Barnett, M., & Squire, K. (2002). Developing an empirical account of a community of practice: Characterizing the essential tensions. *Journal of the Learning Sciences, 11*(4), 489-542.

Barab, S. A., & Duffy, T. M. (2000). From practice fields to communities of practice. In D. H. Jonassen & S. M. Land (Eds.), *Theoretical foundations of learning environments*. Mahwah, New Jersey: Lawrence Erlbaum associates, 25-55.

Bennett, C. (2009). Student-Authored wiki textbook in CS1. *Journal of Computing Sciences in Colleges, 24*(6), 50-56.

Chang, C. K., Chen, G. D., & Li, L. Y. (2008). Constructing a community of practice to improve coursework activity. *Computers & Education, 50*(1), 235-247.

Choy, S. O., & Ng, K. C. (2007). Implementing wiki software for supplementing online learning. *Australasian Journal of Educational Technology & Society, 23*(2), 209-226.

Cole, M. (2009). Using Wiki technology to support student engagement: Lessons from the trenches. *Computers & Education, 52*(1), 141-146.

Elgort, I., Smith, A. G., & Toland, J. (2008). Is wiki an effective platform for group course work? *Australasian Journal of Educational Technology, 24*(2), 195-210.

Guzdial, M., Rick, J., & Kehoe, C. (2001). Beyond adoption to invention: Teacher-created collaborative activities in higher education. *Journal of the Learning Sciences, 10*(3), 265-279.

Hamer, J., Purchase, H. C., Denny, P., & Luxton-Reilly, A. (2009). *Quality of peer assessment in CS1*. Paper presented at the fifth international workshop on Computing education research workshop.

Henri, F., & Pudelko, B. (2003). Understanding and analysing activity and learning in virtual communities. *Journal of Computer Assisted Learning, 19*(4), 474-487.

Hsieha, I.-L. G., & Jr., H. F. O. N. (2002). Types of feedback in a computer-based collaborative problem-solving group task. *Computers in Human Behavior, 18*(6), 699–715.

Johnson, D. W. (2003). Social Interdependence: Interrelationships Among Theory, Research, and Practice. *American Psychologist, 58*(11), 934-945.

Johnson, D. W., & Johnson, F. P. (2002). *Joining Together: Group Theory and Group Skills* (8th ed.). Boston: Allyn & Bacon.

Joy, M. (2005). Group projects and the computer science curriculum. *Innovations in Education and Teaching International, 42*(1), 15-25.

Judd, T., Kennedy, G., & Cropper, S. (2010). Using wikis for collaborative learning: Assessing collaboration through contribution. *Australasian Journal of Educational Technology & Society, 26*(3), 341-354.

Kalles, D. (2008). Students working for students on programming courses. *Computers & Education, 50*(1), 91-97.

Karasavvidis, I. (2010). Wiki uses in higher education: exploring barriers to successful implementation. *Interactive Learning Environments, 18*(3), 219-231.

Kölling, M., & Barnes, D. J. (2004). Enhancing Apprentice-Based Learning of Java. *ACM SIGCSE Bulletin 36*(1), 286-290.

Lave, J., & Wenger, E. (1991). *Situated learning: Legitimate peripheral participation*: NewYork:Cambridge University Press.

Li, L.-Y., & Chen, G.-D. (2009). A Coursework Support System for Offering Challenges and Assistance by Analyzing Students'Web Portfolios. *Educational Technology & Society, 12*(2), 205-221.

Raban, R., & Litchfield, A. (2007). Supporting peer assessment of individual contributions in groupwork. *Australasian Journal of Educational Technology & Society, 23*(1), 34-47.

Robins, A., Rountree, J., & Rountree, N. (2003). Learning and Teaching Programming: A Review and Discussion. *Computer Science Education, 13*(6), 137-172.

Sancho-Thomas, P., Fuentes-Fernández, R., & Fernández-Manjón, B. (2009). Learning teamwork skills in university programming courses. *Computers & Education, 53*(2), 517-531.

Strijbos, J. W., Martens, R. L., & Jochems, W. M. G. (2004). Designing for interaction: Six steps to designing computer-supported group-based learning. *Computers & Education, 42*(4), 403-424.

Tharp, R. G., & Gallimore, R. G. (1988). *Rousing minds to life: Teaching,learning, and schooling in social context*. Cambridge: Cambridge University Press.

Vygotsky, L. (1978). *Mind in society: The development of higher psychological processes.*: Cambridge, MA: Harvard University Press.

Weaver, D., Viper, S., Latter, J., & McIntosh, P. C. (2010). Off campus students' experiences collaborating online, using wikis. *Australasian Journal of Educational Technology & Society, 26*(6), 847-860.

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

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

Xu, L. (2007). Project the wiki way: using wiki for computer science course project management. *Journal of Computing Sciences in Colleges, 22*(6), 109-116.

Zeller, A. (2000). Making students read and review code. *ACM SIGCSE Bulletin 32*(3), 89-92.

