Developing an Interactive Practice Tool in PeerSpace for First Year Computer Science Students

Cen Li, Zhijiang Dong, Roland H. Untch, and Divya Jagadeesh

***Abstract*—Addressing the needs of students in entry-level Computer Science courses, an online social-network-based learning environment, PeerSpace, was successfully developed to enhance student learning and performance. PeerSpace integrates a suite of Web 2.0 tools that promote student interactions on course-related topics as well as purely social matters. Part of this suite is a practice tool, Preparation Station, developed to strengthen and reinforce students' grasp of concepts learned in class, to encourage student participation in PeerSpace, and to trigger course-related communication among students. PeerSpace is implemented atop the open-source Elgg social network framework. This paper presents design and implementation details of Preparation Station as well as an assessment of its usefulness and effectiveness.**

***Index Terms*—Computer supported collaborative learning, computer uses in education, online learning tool, CS1 and CS2.**

I. Introduction

Education research provides evidence that interacting with peers fortifies the learning process and makes learning more enjoyable [1]-[2]. However a survey of the culture among students in introductory Computer Science (CS) courses has revealed negative behaviors such as: disdain for working in groups, unwillingness to support or aid others and combativeness towards the opinions of peers [3]. This type of peer learning environment has an adverse effect on student's motivation, persistence, and passion towards the course material. Faced with high dropout and failure rates [4], computer science educators have sought for a variety of methods to improve the situation by developing innovative teaching pedagogies and effective teaching and learning tools and environments.

PeerSpace is an online learning environment developed to enhance student learning by encouraging and facilitating the building of peer support networks among students enrolled in entry level CS courses. Peer networks enable the students to support each other both socially and academically and to deal with common difficulties such as stress and isolation. In addition, peer networks serve as a solid foundation for effective peer collaborative learning. With strong peer support, the students will be more comfortable and willing to share knowledge and experiences, exchange ideas, and seek help [5]. Carefully designed learning activities in *PeerSpace* encourage students to help each other and learn from each other. A combination of these is expected to lead to enhanced student learning.

In order to give students opportunities to reinforce course concepts, encourage student participation in *PeerSpace*, and to trigger course-related communication among students, an online practice tool, *Preparation Station*, was created. This tool helps the students gain additional knowledge on concepts covered in class by working on a set of course-related practice questions. The questions, administered via this component of *PeerSpace*, are selected by the instructor according to the current needs of the class.

Most existing online practice tools have been developed as stand-alone applications [6]-[7]. Others have been integrated into larger learning systems. Some of these systems were developed and distributed commercially while others were developed in-house in order to support specific technology-mediated collaborative learning programs. Key examples in the first category are Blackboard™ [8], ATutor [9] and Sakai [10]. The WebTycho system developed by the University of Maryland [11] is an example of in-house systems. Rossling *et al.* [12] conducted an online survey to determine "the nature of computer science educators' attitudes toward learning management systems". Respondents to the survey claimed that "large scale LMS's were not flexible enough and that they did not cover the entire spectrum of what was required for computer science education" [12]. Respondents also stated that "cooperation and data exchange between LMS and other computer science specific tools was not very good" [12]. From the survey responses the authors concluded that there is a lot of scope for "integrating computer science specific tools with the broader and more general capabilities of LMSs" [12].

The *Preparation Station* tool in the *PeerSpace* environment is designed based on the fundamental ideas of a learning content management system and online social networks. The various sub-components of the *Preparation* Station tool in combination with the social networking features of the *PeerSpace* environment provide various LCMS features such as web-based course content delivery, practice test handling, assessment and report generation, score display and student discussion forums. Also, since *PeerSpace* is developed based on *Elgg*, an open source social networking framework, it is extensible and allows the addition of CS specific functionality through Plugins. The *PeerSpace* environment provides for seamless integration

Manuscript received October 15, 2012; revised December 14, 2012. This work is supported by the CCLI-TUES program of National Science Foundation under grant DUE-0837385 and a grant from the Sponsored Research Program at Middle Tennessee State University.

Cen Li, Zhijiang Dong, and Roland H. Untch are with the Department of Computer Science, Middle Tennessee State University, Murfreesboro, TN 37132 USA (e-mail: Cen.Li@ mtsu.edu, Zhijiang.Dong@mtsu.edu, Roland.Untch@ mtsu.edu).

Divya Jagadeesh was a graduate student in Middle Tennessee State University. She is now with Hospital Corporation of America, Nashville, Tennessee 37203 USA (e-mail: divyajagadeesh@gmail.com).

between the various student collaborative activities and course management features.

We have implemented the *Preparation Station* tool within the *PeerSpace* learning environment. Students may seek help on these exercises from the other students in the learning community through online communication channels such as group discussion forum and online chats. Students may serve as tutors to other students by answering email, or by posting a reply to a question on the course discussion forum. Students will benefit from the convenience and flexibility offered by anytime, anywhere access.

II. PEERSPACE

PeerSpace is developed based on Elgg [13], an open source social networking framework. Elgg comes with a number of basic features, some of which are directly applicable in Peerspace. These include:

- User profile: Students may edit their profile information such as the profile icon, interests, email, and personal web links;
- Friends: Students can request/add other students to be their friends;
- Personal blogs: Students may post blogs with multimedia content and comment on other students' blogs;
- Groups: Users may create groups or join existing groups. The group forum discussion feature allows a private discussion among the group members.

Fig. 1. PeerSpace front page

To make *PeerSpace* an environment that foster a strong sense of community among the freshmen students taking the CS courses, a number of new features have been developed as additional plugin modules and widgets:

- Scoring for *PeerSpace* community contribution to keep track of a student's participation/contribution to the community;
- Groups with high average member contribution scores are shown on the front page of PeerSpace;
- A custom-built forum discussion module that allows hierarchical threading and thumbs up and down voting;
- An online chat in PeerSpace is achieved by running an OpenFire server and the Pidgin clients. The online chat feature is directly connected to Peerspace where the users and groups created within PeerSpace are automatically created as users and groups within the OpenFire database.
- The group wiki plugin has been adapted to allow the study group members to collaboratively creating/editing wiki documents.

Once logged in, a student is presented with the PeerSpace front page. The *PeerSpace* front page is designed to show the current community status, e.g., latest blog and forum posts, online users, list of friends, group membership; as well as the current rankings of the top students and top groups. Fig. 1 shows a snapshot of the *PeerSpace* front page. Each component in the front page (enclosed in a shaded rectangle) is built as an *Elgg* widget. Students may customize the features of individual widgets as well as the layout of the front page with an easy drag-and-drop interface.

III. PREPARATION STATION

The Preparation Station tool is a new component of PeerSpace designed to help the students better understand the course material by working on additional course related exercises outside of class. The exercises are designed by the instructor according to the current needs of the class. Each exercise is a collection of questions on the same topic, i.e., array. In addition to helping the students reinforce the material taught in class, the exercises are designed to trigger course related discussion among students on PeerSpace leading to more peer communication, collaboration and peer tutoring. The Preparation Station tool has been designed such that it is convenient for an instructor to:

- create, edit, organize, and view the questions;
- create exercises using subsets of the questions, and order the questions within each exercise; and
- create and edit assignments for the intended student groups and with appropriate due dates.

A student can start and stop working on the assignment at any time and resume later. When student resumes the assignment, the questions start from the point where the assignment was left off. Students will receive immediate feedback on the correctness of the answer as well as an explanation to the question if the answer is incorrect. Information about assignment progress, for example total number of questions answered, total correct answers and number of questions remaining, is available for the students at any time. If the instructor wishes to allow the students to work on the questions more than once, a reset option is available for the students to work through the questions again.

A. *Create / Edit / Delete Questions*

Separate templates for creation of multiple choice questions and true/false questions are designed. The multiple choice question template has provision for entering the question, its choices, the correct answer, a small explanation of the concept behind the questions for students, comments for instructor reference, the difficulty level of the question and a provision to choose a keyword from an existing list of keywords to indicate the concept covered by the question. The true/false question template has a similar format. Instead of the fields for the possible choices of a question, a field is provided for the correct answer. Preview of the question is made available in the template pages to allow the user to

preview the questions while creating them. The questions used in *Preparation Station* are taken mostly from the test bank questions supplied with the textbook by Dale [14]. Fig. 2 shows the template designed for the creation of a multiple-choice question.

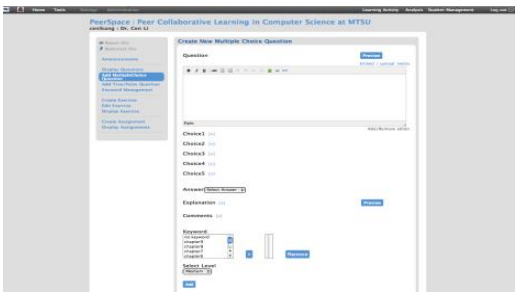

Fig. 2. A template for creating multiple-choice questions

B. *Create / Edit / Delete Exercises*

An exercise contains a set of questions covering one or more concepts. Each exercise has a title and is linked to one or more existing keywords. Linking an exercise to a keyword includes all the questions associated with that keyword as part of that exercise. In addition, the instructor may choose to include or exclude certain questions for the exercise depending on the progress of a class.

Although all the questions associated with the keywords chosen for an exercise get included in the exercise, the questions are all *disabled* by default at first. An instructor can enable or disable all the questions, or only a selected number of questions for the exercise. The instructor is also able to order the questions selected for the exercise by drag and drop the question to the desired position. Deletion of an exercise also deletes the details of all the assignments that are linked to the exercise. Exercise deletion is not allowed if an exercise is linked to a current assignment, and the students have been working on the assignment.

C. *Create / Edit / Delete Assignments*

An assignment assigns an exercise to one or more groups of students. Each assignment is given a submission deadline. The students in the selected groups will be able to work on the assignment till the specified deadline. A mechanism is built-in for the instructor to specify whether (s)he wishes to allow or disallow the students to reset their work on the assignment.

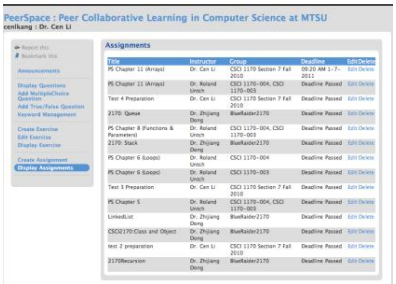

Fig. 3. An instructor's view of all the exercises created

Fig. 3 shows a list of assignments created for various classes during Fall 2010. Click on the "edit" button of an assignment leads to a page for editing the title, student groups, exercise, as well as the deadline of the assignment. Click on the "delete" button deletes all information relates to that assignment. To prevent accidental deletion of students' work on their assignments, information about the students' progress on an assignment is displayed before a confirmation is requested for deletion.

D. *Student Performance Analysis*

The Assignment Analysis module helps the instructors assess the student performance on the assignments. Student progress information such as total number of questions answered and total correct number of questions answered are recorded for all the assignments. The Analysis-By-Student option allows an instructor to view one student's performance on a select exercise. The Analysis-by-Group option allows an instructor to view a cumulative report of the performance of all the students in one group on all the exercises the group worked on. The report lists all the assignments that the group has worked on during the specified time period. For each assignment, the report shows the ids of the questions in the assignment, the total number of students who gave the correct answer, as well as the distribution of the student responses across different answer choices. The Analysis-by-Exercise option allows an instructor to view the performance of all the student groups that have worked on one chosen exercise in a given time period. In the report, for each group, the ids of the questions in the assignment, the total number of students who gave the correct answer to the question, as well as the distribution of the student responses across different answer choices are displayed. Fig. 4 shows an example of Analysis-by-Group for one CS1 class group.

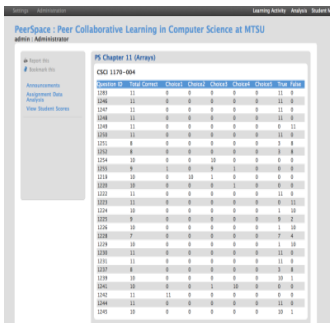

Fig. 4. An example report summarizing the performance of a group on an assignment

IV. Implementation

A. *The Elgg Framework*

PeerSpace is developed based on *Elgg* [13], an open source social networking framework which provides the basic functionality to run a social network. *Elgg* is extensible which allows it to be customized with different features through the use of plugin modules.

Elgg is primarily implemented using PHP scripting language and supports handling of the backend database using MySQL. The main challenge encountered with development using *Elgg* is, by design, creation of one's own database tables is not recommended/allowed. This is because, "*Elgg* is underpinned by a flexible, generic data model" due to which plugins and custom code need not be updated "if

new database functionality is introduced, or if a new data back-end is inserted to support multiple servers or other infrastructure requirements" [13].

B. The Elgg Data Model

Elgg is running on a unified data model, based on atomic units of data called entities. The top level class is called ElggEntity, which defines for all entities the most common properties and behaviors such as GUID (Globally Unique Identifier) to identify a ElggEntity object uniquely, access permissions to control object access, an arbitrary subtype to group similar entities, an owner, and the site that the entity belongs to. ElggEntity has four main specializations, which provide extra properties and methods to handle different kinds of data: *ElggObject* to manage objects like blog posts, uploaded files, questions, assignments, and exercises; *ElggUser* to manage users in the system; *ElggSite* for each site within an Elgg install; and *ElggGroup* to manage user groups.

Entities can be extended with extra information using Metadata and Annotations. Metadata "is information that is added to an object to describe it further" [13]. Examples of metadata include question type, assignment deadline, and number of questions answered. Annotations "is the information generally added by third parties to the information provided by the entity" [13]. For example, comments and ratings of a blog post are both annotations.

In order to make the system developed using Elgg more stable, plugin modules are strongly discouraged from dealing with database directly. Any access to the database should be done using the predefined functions available to handle objects in Elgg. Such an approach allows for content created by different plugins to be mixed together in consistent ways [4].

C. Implementation of the Student Interface

Each assignment in the *PeerSpace* environment is stored as an ElggObject of subtype "assignment". In addition to the predefined properties of an ElggObject such as GUID, access permissions, owner, and title, the following set of metadata have been created for each assignment entity:

- *Groupid* – an array of guids of all the groups to which the assignment is assigned to;
- *Exeid* – the guid of the exercise that is used in the assignment;
- *Deadline* – the deadline date for the assignment;
- *Resetoption* – set to 0 if students are not allowed to work multiple times on the assignment, and to 1 otherwise; and
- *Questionguid* – an array of guids of all the enabled questions in the assignment.

A student is allowed to view only the assignments that are assigned to him, i.e., assignments assigned to the groups in which the student is a member of. The assignment display screen (similar to that of Fig. 3) displays the title and deadline for each of the retrieved assignment using the assignment object's "title" property and "deadline" metadata. A "Reset" button is also displayed alongside the title if the "resetoption" metadata of the assignment is set to 1. Otherwise, a "not allowed" message is displayed in that column.

When a student first works on an assignment, an ElggObject is created to hold the student's assignment progress and performance details. This ElggObject has subtype "testdetails". Metadata are created for this object to record the GUID of the user; the GUID of the assignment; the GUID of the exercise the assignment is linked to; the total number of questions answered; the percentage of the questions answered correctly; the ids of the questions in the order answered by the student; as well as the student answers.

When a student starts working on an assignment, the assignment GUID is used to retrieve the corresponding assignment object. Questions in the assignment are retrieved using the "questionguid" metadata of the "assignment" object and displayed one at a time to the student. Fig. 5. shows how the questions are presented to the students one at a time.

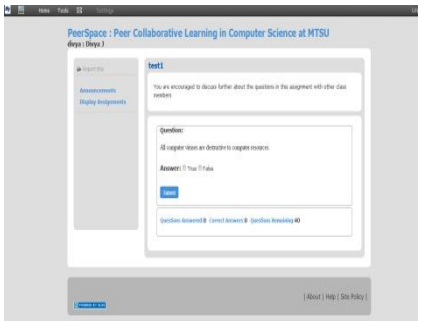

Fig. 5. The exercises are presented to students one question at a time in preparation station

Questions are stored as ElggObjects of subtype "Question" with metadata created for information including the answer for the question; the question choices; the question type, as well as the answer explanation. The question choices and the assignment progress information are populated using the metadata of the "question" and "testdetails" objects respectively. When a student submits an answer, the student's answer is verified against the "answer" metadata of the "question" object and the metadata values for the student's "testdetails" object are updated accordingly. The feedbacks for the student's answer are also updated from the corresponding metadata of the "question" object.

A student can exit the assignment at any time. When a student resumes an assignment, the "testdetails" object created for the student is retrieved based on the assignment's GUID and the student user's GUID. When a student reset his assignment, the "testdetails" object of the student for the assignment is retrieved. Then the "testdetails" object is deleted.

V. Experimental Analysis

PeerSpace has been used by CS1 and CS2 students in the Spring and Fall 2010 semesters. In each semester, two CS1 sections were designated the Experiment and the Control group at the beginning of the semester. The two sections in a semester were taught by the same instructor with the same lecture material and textbook, albeit the instructors participated in the study in the Spring and the Fall semester are two different person. In both semesters, the *Preparation*

Station tool was used mainly as a test preparation tool with assignments given to the students prior to tests.

When analyzing the aggregated results from two groups of students from these two semesters, the two-tailed t-test was used to compute the statistical significance between the mean values obtained from the two groups. If the test scores of the group of students who used the practice tool differs from that of the group of students who did not use to the tool, the test helps to answer the question weather any difference observed is statistically significant.

Table I summarizes the comparison results between the control group and the experiment group from the two semesters. In the Spring 2010 semester, the mean score for the students in the experiment group for both Test 3 and Final Exam, 33.1 (out of 50 points) and 75.1, are higher than those from the students in the control group, 29.3 and 72 respectively. Similarly, in Fall 2010, the mean score for the students in the experiment group for Test 2 and Test 3, 75.4 and 62.4, are higher than those from the students in the control group, 70.7 and 59.8. The last column of the table presents the t-test results of these comparisons. The t-test results suggest that the higher mean scores observed from all four tests for the experiment group are not statistically significant with r-values of 0.296, 0.656, 0.45 and 0.76 respectively.

TABLE I: COMPARISONS OF THE CONTROL AND EXPERIMENT GROUP TEST SCORES FROM 2010 SPRING AND FALL SEMESTER

	Tests	Group	Sample Size	Mean (std dev)	r-value
Spring 2010	Test 3	Control	18	29.3(11.1)	0.29
		Experiment	21	33.1(11.0)	
	Final Exam	Control	16	72(18.5)	0.65
		Experiment	17	75.1(19.7)	
Fall 2010	Test 2	Control	21	70.7(24.8)	0.45
		Experiment	26	75.4(17.8)	
	Test 3	Control	18	59.8(29.4)	0.76
		Experiment	24	62.4(26)	

Student surveys have been conducted at the end of each semester to gather student opinions about the usefulness of *Preparation Station* in learning the course material and preparing for tests. Fig. 6 summaries the results. For the statement "the *Preparation Station* tool helped me in understanding the course material better", a combined 63% of the students gave positive opinion, with 30% of these students strongly agreeing with the statement (Fig. 6 (a)). For the statement "after using the *Preparation Station* tool, I feel more confident about taking a test on this material", a similar trend is observed with 28% of the students strongly agreeing and another 31% of the students agreeing (Fig. 6 (b)). When asked their opinion about the social aspect of the tool, "the preparation station tool benefits me more since I can discuss the solutions and clarify concepts with my classmates when working through the questions", 23% of the students strongly agree the statement and another 30% of the students agreeing. These results show that a majority of the students found *Preparation Station* tool to be a useful tool for learning course material and preparing for the tests. It is also observed that for each survey statement, the answers from approximately 30% of the students show neutral or negative responses. When examining the student performance data in *Preparation Station*, it was found that many of the students who did not answer positively in the survey did not work on or complete the *Preparation Station* assignments, mainly due to the fact that the *Preparation Station* assignments are given as voluntary assignments. For example in Spring 2010, 30% of the students fully utilized the *Preparation Station* tool and 65% used it partially. The *Preparation Station* usage is higher in the Fall 2010 semester. It is understandable that the students cannot realize the benefits of the tool unless they use it.

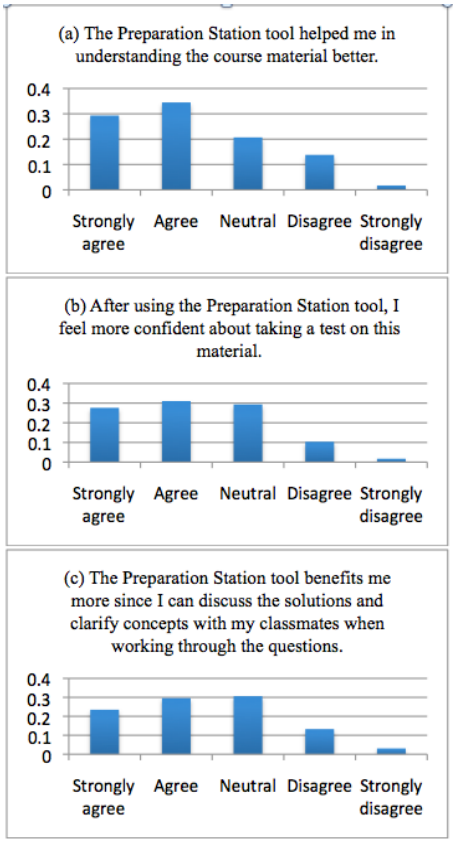

Fig. 6. Student survey results concerning the usage and usefulness of preparation station for learning in CS1

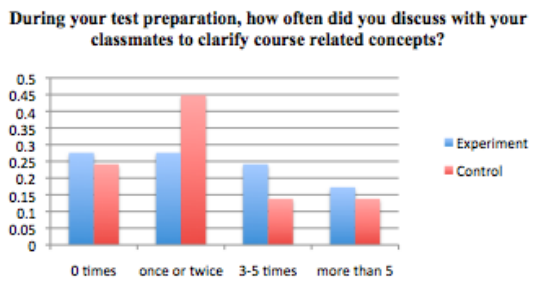

Fig. 7. Comparing the experiment and control groups on the amount of learning occurred during studying

A secondary objective of implementing the *Preparation Station* tool within PeerSpace is to promote peer learning. Students having questions while working through the *Preparation Station* questions can ask for help in PeerSpace

through public (PeerSpace site-wide) or class group discussion forum, chat online with fellow students, or talk to them in person. Figure 7 presents the comparison results between the experiment and the control group students in terms of the amount of peer learning occurred during test preparation. It can be seen that the percentage of students who has consulted with their peers for 3-5 times or for more than 5 times during studying is higher among the experiment group students than among the control group students. There is also a slightly higher percentage of students never discussed course related concepts with their classmates during test preparation. One explanation may be because if one answers a *Preparation Station* question incorrectly, a detailed explanation about that question is always presented, which maybe sufficient for many students to clear their misconception, therefore no need to consult a fellow student. Additional investigation is needed to fully understand this data.

VI. Conclusions

In this paper, the motivation, design, and implementation of an online practice tool developed atop the Elgg social network engine is presented. In addition, comparative experiments have been conducted with CS1 students in two semesters. Overall from the student survey responses, it can be seen that the students did perceive *Preparation Station* as a good and useful learning tool, especially those who made full use of the tool. For the Experiment and the Control groups of students, the mean test scores of the students in the Experiment group was consistently higher than the mean test scores of the students in the Control group (although variations in individual performance rendered this difference as not statistically significant.) Although performance improvement have been observed in the students who worked on *Preparation Station* exercises for test preparation, the improvement in student performance in a whole was not up to the level we had initially anticipated. The benefits of the *Preparation Station* tool on student learning are highly dependent on the level of participation. In the future experimentation, it is essential that the *Preparation Station* assignment be made mandatory, at least initially, so that the students have a chance to use it and realize for themselves that it is helpful.

References

[1] A. Perez-Prado and M. Hirunarayanan, "A qualitative comparison of online and classroom-based sections of a course: Exploring student perspectives," *Educational Media International*, vol. 39, no. 2, pp. 195-202, 2002.

[2] K. Swan, "Building learning communities in online courses: the importance of interaction," *Education Communication and Information*, vol. 2, no. 1, pp. 23-49, 2002.

[1] W. M. Waite and P. M. Leonardi, "Student culture vs. group work in computer science," in *Proc. 35th SIGCSE Technical Symposium on Computer Science Education*, Norfolk, VA, USA, March 2004, pp. 12-16.

[2] P. Kinnunen and L. Malmi, "Why students drop out CS1 course?" in *Proc. of the 2nd International Workshop on Computing Education Research*, Canterbury, UK, Sept. 2006, pp. 97-108.

[3] V. Tinto, "Classroom as communities: exploring the educational character of student persistence," *Journal of Higher Education*, vol. 68, no. 6, pp. 599-623, 1997.

[4] J. Yoo, C. Pettey, S. Yoo, J. Hankins, C. Li, and S. Seo, "Intelligent Tutoring System for CS-I and II Laboratory," in *Proc. 44th ACM Southeast Conference*, Melbourne, FL, March 10-12th, 2006, pp. 186-191.

[5] J. Yoo, C. Li, and C. Pettey, "Adaptive teaching strategy for online learning," in *Proc. International Conference on User Interface*, San Diego, CA, Jan. 2005, pp. 266-268.

[6] Blackboard™. (November 20th, 2012). [Online]. Available: http://www.blackboard.com.

[7] Atutor. (November 20th, 2012). [Online]. Available: http://www.atutor.ca. Accessed

[8] Sakai. (July 27 2012). [Online]. Available: http://www.sakaiproject.org. Accessed

[9] WebTycho. (July 27 2012). [Online]. Available: http://www.makli.com/webtycho-0035823. Accessed

[10] G. Rossling, L. Malmi, and M. Clancy, "Enhancing learning management systems to better support computer science education," *ACM SIGCSE Bulletin*, vol. 40, no. 4, pp. 142-166, 2008.

[11] N. Dale and C. Weems, *Programming and problem solving with C++*, Sudbury, MA: Jones and Bartlett Publishers, 2010.

[12] Elgg: (November 20th, 2012). A Powerful Social Engine. [Online]. Available: http://www.elgg.org.

Cen Li obtained a M.S. and a Ph.D. in computer science from Vanderbilt University (Nashville, TN, USA) in 1995 and 2000 respectively. She has a B.S. degree in computer science from Middle Tennessee State University (Murfreesboro, TN, USA) 1993. Dr. Li is currently a Professor in the Department of Computer Science at the Middle Tennessee State University. Her main research interests are machine learning, data mining, information retrieval and robotics. Dr. Li is a member of the ACM.

Zhijiang Dong obtained Ph.D. in Computer Science from Florida International University (Miami, FL, USA) in 2006, M.S. in Computer Science and B.S. in Applied Mathematics from Huazhong University of Science and Technology (China) in 1997 and 1994 respectively. Dr. Dong's research interests include software engineer, formal methods, verification and validation. Dr. Dong is currently an Associate Professor in the Department of Computer Science at Middle Tennessee State University. Dr. Dong is a member of the ACM.

Roland H. Untch has a Ph.D. in computer science from Clemson University (Clemson, South Carolina) 1995, an M.S. in computer science from DePaul University (Chicago, Illinois) 1983, and a B.A. in management from Mundelein College/Loyola University of Chicago (Chicago, Illinois) 1979. He is currently a Professor in the computer science department of Middle Tennessee State University. Dr. Untch is a member of the ACM and the IEEE Computer Society.

Divya Jagadeesh was born in January 1, 1985, in Chennai, India. She graduated with a Bachelor of Science in Computer Science degree from Anna University, Chennai, India in 2006 and a Master of Science in Computer Science degree from Middle Tennessee State University, Murfreesboro, Tennessee, USA in 2010.

She worked as a Programmer Analyst in Cognizant Technology Solutions from August, 2006 to December, 2007. She also worked as a Research Assistant under the guidance of Dr. Cen Li in Middle Tennessee State University from June, 2009 to May, 2010. She currently works as a Senior Report Solutions Analyst in Hospital Corporation of America, Nashville, Tennessee. Her current and previous