A Coursework Support System for Offering Challenges and Assistance by Analyzing Students’ Web Portfolios

Liang-Yi Li and Gwo-Dong Chen
Department of Computer Science and Information Engineering, National Central University, Taiwan // Tel: +886-3-4227151 ext. 35327 // Fax: +886-3-4273485 // liang@db.csie.ncu.edu.tw // chen@csie.ncu.edu.tw

ABSTRACT
Students can practice skills and acquire knowledge by doing coursework. However, in conventional coursework activities, each student is assigned the same exercises, without considering learners’ diversity. Moreover, students typically have difficulty in receiving assistance for completing their exercises after class. Therefore, some students cannot learn effectively when doing their coursework. This work presents a Web Coursework Support System (WCSS) to assist students doing coursework within their Zone of Proximal Development (ZPD). This system applies decision tree analysis methodology for selecting programming exercises at a level of difficulty suited to each student. The assigned exercises thus challenge students. To assist students in completing these assigned exercises, this system allows students to access distributed online learning materials related to a programming concept through links on a single web page and motivates students to answer posted questions in a discussion forum. Experimental results show that the exercises-completed rate and the rate of the assigned exercises completed with assistance were increased. They indicate that WCSS can increase likelihood that students do coursework within their ZPD by offering challenges and assistance. Furthermore, some students’ responses were insightful in understanding the benefits and limitations of this system.

Keywords
Coursework, Concept map, Decision tree, Zone of Proximal Development

Introduction
Coursework is a significant learning activity to supplement classroom instruction (Cosden, Morrison, Albanese, & Macias, 2001). In conventional coursework activity, the teacher first assigns several exercises to all students. The students then do the exercises and hand them in. Finally, the teacher comments on each student’s work and credits each student with a grade. In the process, students can construct knowledge by reading textbooks, discussing topics with peers, and making inquiries through online resources (Cooper & Valentine, 2001; Epstein & Van Voorhis, 2001). A teacher can also evaluate the learning performance of students and understand their learning status based on the quality of their coursework.

However, the teacher assigns the same exercises to a diverse group of students who have varying learning statuses. Excellent students may feel that the exercises are too easy to teach them anything, while below-average students may feel that the exercises are too hard to allow them to learn (Corno, 2000). Additionally, learning aids, such as libraries and capable classroom peers, are located in different places. Students generally have difficulty receiving these aids to complete their exercises after class (Glazer & Williams, 2001). In these situations, students often fail to finish their coursework, or, occasionally, plagiarize.

Recently, some online assignment systems have been designed to support students and teachers in a conventional coursework activity. For instance, some systems provide assistance for teachers and students to manage the process of the conventional coursework activities, such as automatic assignment submission, assessment, and feedback (Collis, De Boer, & Slotman, 2001; Dawson-Howe, 1996; Lee & Heyworth, 2000; Saikkonen, Malmi, & Korhonen, 2001). The systems can help teachers manage the process of an assignment, and so reduce teachers’ workloads. However, they do not provide support for the assigning of appropriate exercises for each student or for students’ completion of these assigned exercises in the coursework activity.

Some systems provide personal tutoring that assigns adaptive questions for students and then guides students of varied abilities to correct their own assignment errors (Lilley, Barker, & Britton, 2004; Murray & Arroyo, 2002; Syang & Dale, 1993). These systems usually are applied in the Computerized Adaptive Test (CAT) domain to select the most appropriate questions based on Item Response Theory (IRT) (Lord, 1980; Rogers, Swaminathan, & Hambleton, 1991). However, in order to achieve reliable results, these systems require substantial interaction.
between a user and the system. Additionally, IRT is most suited to multiple choice or fill-in-the-blank questions and is less effective for open-ended questions, like programming exercises that need students to write a program to solve a problem (e.g., sum the odd integers between 1 and 99).

This study develops a Web Coursework Support System (WCSS), which is designed for supporting students who are learning the Java programming language in the coursework activity. It is composed of three subsystems: a Personalized Assignment Dispatching System (PADS), an e-dictionary system, and a peer recommendation system. The design of WCSS is based on the work of Vygotsky, who proposed Zone of Proximal Development (ZPD) as “the distance between the actual developmental level as determined by independent problem solving and the level of potential development as determined through problem solving under adult guidance or in collaboration with more capable peers” (Vygotsky, 1978, p. 86). Tharp and Gallimore (1988) proposed a four-stage ZPD model. In their model, ZPD is created when assistance is offered by more capable others or by the self-guidance. After learners progress through the ZPD, the skills and knowledge used within ZPD are internalized and fully developed. Therefore, teaching and learning occur only when assistance is provided to learners in their ZPD. Such assistance consists of six means of assistance: modelling, contingency managing, feeding back, instructing, questioning, and cognitive structuring (Tharp & Gallimore, 1988).

The WCSS provides three ways of helping students performing coursework within their ZPD. First, the programming exercises selected by PADS must be at the appropriate level of difficulty (this study views an exercise as being at the appropriate level of difficulty for a student only if it can be completed by the student under the assistance provided). The PADS analyzes students’ web portfolios to understand their knowledge status. According to the result of the analysis, the PADS select programming exercises that students potentially have difficulty in completing. Thus, these selected exercises can be seen as questioning assistance that asks students to apply and synthesize some programming concepts that they are not proficient now to complete their exercises. Second, the e-dictionary system that organizes useful learning materials with a concept map is designed to provide support for students to learn programming concepts related to these exercises. The concept map is used to help students understand the overview of Java programming and the relationship between different concepts before they select detailed elements to study. The learning materials associated with a concept provide a detailed explanation and related examples. Students can understand the meaning of a concept through its explanation and observe and imitate the related examples to learn how to apply this concept in a program (Astrachan & Reed, 1995; Schworm & Renkl, 2006). Thus, the e-dictionary system can be treated as a modeling and cognitive structuring assistance to help students completing these selected exercises independently. If students still cannot complete these exercises, then they can post their questions on the discussion forum. Third, the peer recommendation system designates two capable classroom peers for each posted question and employs some strategies to motivate them to resolve the questions with the questioner. The peer recommendation system can be seen as a contingency managing assistance that encourages peers to resolve the posted questions. In addition, the recommended peers can be seen as a providing assistance by supporting students completing these programming exercises collaboratively. Ideally, students should do their coursework within their ZPD through challenging them and scaffolding them. Table 1 presents an overview of the design issues for WCSS. The first and second columns of Table 1 list the requirements and related reasons at the time that the system was designed. The third column of Table 1 indicates how to meet those requirements in the proposed system.

<table>
<thead>
<tr>
<th>Requirements</th>
<th>Reasons</th>
<th>Solutions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Assigning programming exercises that have a level of difficulty suited to each student</td>
<td>Assistance provided for learning through challenging engagement</td>
<td>PADS</td>
</tr>
<tr>
<td>Accessing web learning materials through links on a single web page</td>
<td>Assistance provided for completing exercises independently</td>
<td>E-dictionary system</td>
</tr>
<tr>
<td>Motivating capable classroom peers to answer assigned questions</td>
<td>Assistance provided for completing exercises collaboratively</td>
<td>Peer recommendation system</td>
</tr>
</tbody>
</table>

Table 1. Relationship between proposed solutions and requirements

System Overview

In this study, students learned in a blended learning environment. That is, a teacher lectured for three hours weekly, and students took an exam every two weeks in the classroom. Students also learned via a web-based learning system after class. In the web-based learning system, students could discuss items on a discussion forum, submit coursework
assignments on the WCSS, access learning materials, including completed assignments and lecture slides, and perform keyword self-assessments. Some tasks, such as weekly keyword self-assessments, completing coursework assignments every two weeks, and taking an exam every two weeks, must be performed periodically.

All students’ web portfolios can be categorized as three types: learning products, learning behavior, and learning performance (Table 2). An e-dictionary system links these learning products with a concept map to allow students access learning materials related to a concept through links on a single web page. Additionally, to generate pedagogical rules for selecting appropriate programming exercises and suitable capable classroom peers, learning behavior and learning performance were employed to construct a student model. The student model includes students’ learning status for every programming concept in the course and learning performance in different learning activities. Based on the student model, the PADS uses a decision tree analysis methodology to generate a decision model that can determine whether a programming exercise is at the appropriate level of difficulty for a student. Moreover, the peer recommendation system is used to select two capable classroom peers for each posted question based on the student model. Figure 1 illustrates the system architecture of the web-based learning system.

Table 2. Three web portfolio types

<table>
<thead>
<tr>
<th>Category</th>
<th>Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>Learning product</td>
<td>Discussion threads, completed assignments</td>
</tr>
<tr>
<td>Learning behaviour</td>
<td>User logins, read /post /reply to a discussion topic, and submit /review /read completed assignments.</td>
</tr>
<tr>
<td>Learning performance</td>
<td>Exam grades, results of keyword self-assessments and coursework grades.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Students</th>
<th>(Web Browser)</th>
</tr>
</thead>
</table>

Web-Based Learning System (IIS Web Server)

<table>
<thead>
<tr>
<th>E-book</th>
<th>Discussion Forum</th>
<th>Coursework Submission and Evaluation</th>
<th>Self-assessment</th>
</tr>
</thead>
</table>

Web Coursework Support System

<table>
<thead>
<tr>
<th>E-dictionary</th>
<th>PADS</th>
<th>Peer Recommendation System</th>
</tr>
</thead>
<tbody>
<tr>
<td>Concept Map</td>
<td>Decision Model</td>
<td>Student Model</td>
</tr>
</tbody>
</table>

Web Portfolios (Oracle Database System)

Figure 1. System architecture of the web-based learning system

Concept mapping is a technique that is widely used to represent knowledge in a graphical format (Jonassen, Beissner, & Yacci, 1993; Novak & Gowin, 1984). A concept map is a node-link diagram, which represents a concept using a node and a relationship (for example, “is-a,” “related-to,” or “part-of”) using a link (Novak, 1991). Figure 2 shows a part of a concept map drawn by a teacher, representing partial object-oriented programming knowledge in a Java programming course.
Concept mapping has been used for a variety of educational purposes, such as assessing learner's knowledge status (Hoeft et al., 2003; Turns, Atman, & Adams, 2000), sharing meaning between people (Chiu, 2004; Hughes & Hay, 2001), planning the process of problem-solving (Chang, Sung, & Lee, 2003), organizing knowledge (Chang, 2002; Lee, 2004; Triantafillou, Pomportsis, Demetriadis, & Georgiadou, 2004), and representing a learner's knowledge structure (Kuo, Lien, Chang, & Heh, 2004). In this study, a concept map drawn by teachers was applied to present students' knowledge structure in Java programming and to organize learning products.

Each student has his/her own concept map to present the student's learning status for every programming concept in the course. Each concept node in a student's concept map has a value representing the student's proficiency level in the concept. The value for each concept node is generated from the results of exams and keyword self-assessments using a formula. Additionally, learning products are connected to related concepts in a concept map to provide learning scaffolds. Three tables are used to store the concept map (table “Concept_Map” in Figure 3), students' learning status for every programming concept (table “Learning_Status” in Figure 3), and the relationship between the concept map and those learning products (table “Learning_Material” in Figure 3). Figure 3 presents the three table schema and their relationship.

**Figure 2. A concept map representing partial object-oriented programming knowledge**

**Operations and interface of WCSS**

The teacher and Teacher Assistants (TAs) design ten candidate programming exercises based on the syllabus every two weeks. The PADS selects three exercises from these candidate exercises for each student, primarily based on the decision model in the PADS. A web page displaying three hyperlinks representing the three selected exercises is
shown when a student logs into the WCSS. The student can click one of the three hyperlinks to link to a respective exercise web page.

An exercise web page (Figure 4) consists of four sections: problem description, concept, discussion, and submission. A student receiving an exercise first must understand the programming problem and then plan the program based on the problem description. If the student does not know how to solve the problem, then the concepts shown on the concept section can remind the student which concepts should be adopted to solve the problem. Students who are not familiar with these concepts can learn about them on the e-dictionary system through the hyperlinks in the concept section. Students who still cannot solve the problem after using the e-dictionary system can post questions on the discussion forum via the hyperlink in the discussion section. Meanwhile, two capable classroom peers chosen for each question by the peer recommendation system are notified to resolve the assigned questions together with the questioner. Ideally, the student should be able to complete the exercise through this process. Furthermore, the exercise web page presents hyperlinks to the e-dictionary and peer recommendation system, giving students access to assistance, including learning products distributed in different web pages and online capable classroom peers, through links on a single web page.

![Figure 4. The layout of an exercise web page](image)

**Personalized assignment dispatching system**

The PADS applies decision tree analysis methodology to construct a decision model that can determine whether a programming exercise has an appropriate difficulty level for a student. The decision tree analysis originates from the machine learning discipline, which can induce rules for accurately predicting a target attribute from feature attributes.

C5.0 is a machine learning tool for generating decision trees and was developed by Quinlan (Quinlan, 1996). A trial version is available on the Internet at http://www.rulequest.com. The induction performed by C5.0 is based on the value of entropy (Mitchell, 1997). For instance, programming exercises can be clustered into a set of groups based on the values of feature attributes. If programming exercises in a cluster have the same value of the target attribute, then the cluster has the lowest entropy, while if every programming exercise in a cluster has a different value of the target attribute, then the class has the highest entropy. The decision tree tool is used to generate a tree with the minimum entropy.
C5.0 can help teachers to generate a decision tree from the observed data. The procedure used by C5.0 in generating a decision model comprises several serial steps. First, the feature attributes and the target attribute must be selected. Second, teachers must determine how to measure each attribute. Third, the teachers collect observed data composed of a set of instances, described by a fixed set of attributes and their values. Finally, the observed data are input into C5.0, and a decision model is generated. The detailed steps follow.

Determine attributes and measure the level of the attributes

The first step in generating a decision model is to determine the feature attributes and a target attribute. The feature attributes adopted in the decision tree analysis methodology should be associated with the target attribute. The target attribute in this case is whether a programming exercise is suitably difficult for a student. Specific attributes are applied to evaluate the difficulty level of a problem in some studies. For example, Lee and Heyworth (2000) evaluated the complexity of a logarithm problem to a student according to the number of steps needed by the computer system to solve the problem, the number of operators (+, −, *, /, etc.) in the problem expression and the degree of familiarity of the problem to the student; Beck et al (1997) used the proficiency of sub-concepts in a problem to determine its difficulty level. Similarly, to solve a programming problem, a student must be familiar with the related programming concepts and possess an appropriate level of ability in algorithm analysis. Thus, a programming exercise that is suitably difficult for a student may be of a higher algorithm or programming concept complexity level than one that the student can complete without any assistance, but it can be completed with assistance. Therefore, this work selected five attributes: the student’s proficiency level in main concepts used for completing the exercise, the student’s proficiency level in sub-concepts used for completing the exercise, the complexity level in the algorithm of the exercise, the number of lines in program code used for completing the exercise, and the student’s proficiency level in algorithm analysis, as feature attributes. In addition, two other attributes, the coursework grade of the student’s last assignment and login times in the web-based learning system during the last two weeks, were also selected as feature attributes. These two attributes were chosen because the prior coursework performance and the learning motivation (We assume that students performing more login times may associate with that they have a higher learning motivation in learning) may be both related to the target attribute. For example, students having greater learning motivation are willing to spend more time and efforts to solve a more difficult problem. Additionally, students with high prior coursework grade may understand more strategies and knowledge in solving a difficult programming problem. Therefore, this work used seven feature attributes to generate a decision model.

Nine experts (TAs who had learned Java programming for two years and worked as TA for over one year) identified concepts for each exercise. In an exercise, the main concepts are those concepts that are central to the exercise, and sub-concepts are those concepts that have a secondary relationship to it. Item-Objective Consistency (IOC) (Rovinelli & Hambleton, 1977) was used to identify the main concepts and sub-concepts. Figure 5 presents an example that demonstrates how IOC is applied. The X-axis represents the nine experts; the Y-axis represents all concepts that the teacher has instructed in the last two weeks, and the Z-axis represents the ten candidate exercises. Each square contains one of three values, +1, 0, and −1. The value +1 indicates that the expert explicitly confirms that the concept is required by the exercise. The value 0 indicates that the expert non-explicitly confirm the concept demanded by the exercise. The value −1 indicates that the expert can explicitly confirm that the concept is not required by the exercise. The nine experts calculated the IOC for each concept for each exercise after filling in the squares. A concept is a candidate for main concepts when IOC ≥ 0.8 or a candidate for sub-concepts when 0.5 ≤ IOC <0.8. Finally, a panel discussion was conducted by the nine experts to reach a consensus on what were the main concepts and what were the sub-concepts of each exercise. If one of the experts does not agree that a concept is a main concept in an exercise, he/she explains why he/she does not agree and suggests removing it from the candidates for main concepts. If the other experts find the explanation acceptable, they will either remove the concept from the candidates for main concepts, reclassify it as a sub-concept, or discard it entirely.

After identifying the main concepts and sub-concepts for each exercise, a database trigger was then executed to generate the values of two attributes, the student’s proficiency level in main concepts used for completing the exercise and the student’s proficiency level in sub-concepts used for completing the exercise, as one of three values (1 as high, 2 as middle, or 3 as low) for each student for each exercise and to generate the values of three other attributes, the student’s proficiency level in algorithm analysis, the student’s coursework grade in the last assignment, and login times in the web-based learning system in the last two weeks, as one of three values (1 as high,
2 as middle, or 3 as low) for each student according to the student model. The original data type of the five attributes is continuous. Because the decision tree analysis method must sort the data at each step, it may require a large amount of time at sorting the continuous data. In addition, the generated rules may be less meaningful and difficult to interpret. Therefore, we reduce the number of values for each of the five continuous attributes by dividing the range of the values into three intervals based on the method of natural partitioning (Han & Kamber, 2000). The nine experts also measured two attributes, the complexity level in the algorithm of the exercise and the number of lines in program code used for completing the exercise, as one of three values (1 as high, 2 as middle, or 3 as low) for each exercise. The experts measured the value of the complexity level in the algorithm of a exercise according to the criteria: (1) I can design the algorithm for this exercise in one minute; (2) I can design the algorithm for this exercise in five minutes; (3) I have to spend more than five minutes to design the algorithm for this exercise. In addition, the number of lines in program code in each exercise is estimated based on the experts’ experience.

Besides these feature attributes, the value of the target attribute was determined from the answer to a question. Students were required to answer the following question after submitting an assigned programming exercise: “How do you complete the exercise?” The student could choose the following answers: (1) I completed this exercise without any assistance; (2) I completed the exercise with the assistance of related learning materials; (3) I completed the exercise through collaboration with peers; (4) The exercise was far beyond my ability. If the student chose answers 1 or 4, then the system assigned a value of 0 as the target attribute, indicating that the exercise is inappropriate for this student. If the student selected answers 2 or 3, then the system assigned a value of 1 as the target attribute, indicating that the exercise was appropriate for that student.

**Decision model construction**

C5.0 can help teachers to generate a decision tree from the observed data. Observed data from coursework assignments 1–4 were collected to generate a decision model. The data consisted of 230 records. Each record comprised values of eight feature attributes and one target attribute. These records were input into Quinlan’s C5.0 software to generate a decision model. In total, 173 (75%) of the records were used for training, and 57 (25%) were used for testing. The training error rate was 13.3%, and the testing error rate was 21.1%, indicating a 79% probability
that a selected programming exercise was at the appropriate level of difficulty when using the generated decision rules. The decision rules were then programmed. According to the programmed rules, the PADS selected three programming exercises at the appropriate level of difficulty for each student.

**E-dictionary system**

In addition to creating plans, solving problems and making decisions, students learning programming need to understand programming concepts (Jakovljevic, 2003). When computer science students apply their programming knowledge to solve a problem or design a program, programming concepts must be well organized and integrated into their knowledge structures (Murphy et al., 2005). Therefore, the first step in attempting to solve programming problems is understanding programming concepts and organizing these concepts into a knowledge structure.

The World Wide Web (WWW) has become a vast resource for students to acquire knowledge, solve problems, and complete tasks that use web information. Moreover, students can share their learning products and have after-class discussions with peers on the web. The shared learning products present knowledge in a variety of forms. Students can learn programming concepts from different perspectives through these learning products. For example, a discussion thread denotes a process through which students identify and elaborate on ideas and thoughts, debate or interpret their own statements, and provide feedback to other students (Pena-Shaffa & Nicholls, 2004). Similarly, a programming assignment may require a student to provide a narrative description of the required process to complete a programming exercise, the program code of the programming exercise, and explanations on the program code. These learning products are useful for learning, but are distributed among different web pages. Therefore, students have difficulty finding helpful and reliable information (Hill & Hannafin, 2001) and are easily disoriented during the information-seeking process (Marchionini, 1995).

The e-dictionary system organizes useful learning products with a concept map. Each learning product is linked with related programming concepts. Therefore, students just need to find a concept that they want to learn in the e-dictionary system, and then the learning products related to this concept are shown.

In addition, the learning products included in the e-dictionary system were gathered and evaluated by a peer assessment process. The peer assessment process is modelled on the actual journal publication process of an academic society and was conducted last year in the same course (Chang, Chen, & Li, 2008). Every student is endowed with a role (reader, author, reviewer, and editor) according his/her learning performance. Students can submit on any of the following six topics: answers to the assignment, what he/she has learned, summary of discussion, problems, useful websites, and miscellaneousness. Then each submission is evaluated by two reviewers. Finally, an editor summarizes the opinions of all reviewers and determines whether or not the submission will be accepted. If a submission is accepted by editors, it can be included in the e-dictionary system.

The e-dictionary system is designed in two parts: a conceptual structure that denotes how learning products link with a concept map and a user interface that demonstrates how students interact with the e-dictionary system. The conceptual structure is represented as a network structure in which learning products are linked with a concept map. Nodes in the structure are of two types, concept and material. A concept node represents a programming concept in the course, and a material node represents a learning product in the web-based learning system. Each material node can be linked to one or more concept nodes. To link a learning product with the concept map, students must identify which concepts are related to the learning product when submitting the learning product. For example, a student posting a discussion topic or submitting a completed assignment must identify the programming concepts related to the discussion topic or the completed assignment in a listbox. The database schema of the conceptual structure comprises the two tables “Concept_Map” and “Learning_Materials” (Figure 3).

The interface to the e-dictionary system is a collection of web pages accessed by students to learn programming concepts. Figure 6 presents the interface of the e-dictionary system. This interface is divided into four sections. Section 1 is a full-text search engine in which students can search keywords. After a student enters a keyword, Section 2 shows a list of related keywords. If the search engine cannot find this keyword, then the list shows eleven keywords with similar spelling to that keyword. If the search engine can find the keyword, then this keyword is arranged in the middle of this list; and related keywords are arranged in the periphery (see section 2 of Figure 6).
Section 3 has five hyperlinks with this keyword. Table 3 lists the five hyperlinks in detail. The student can click one of the five hyperlinks, and the respective learning materials are then shown in Section 4.

Table 3. The five topics and detailed information

<table>
<thead>
<tr>
<th>Topic</th>
<th>Detailed information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Explanation</td>
<td>The explanation of the subject keyword that is elucidated by teachers</td>
</tr>
<tr>
<td>Example</td>
<td>The completed assignments associated with the subject keyword</td>
</tr>
<tr>
<td>Discussion Thread</td>
<td>The discussion threads associated with the subject keyword</td>
</tr>
<tr>
<td>Related Concept</td>
<td>The keywords linked with the subject keyword in a concept map</td>
</tr>
<tr>
<td>Web search</td>
<td>The search results of the subject keyword from a search engine</td>
</tr>
</tbody>
</table>

**Peer recommendation system**

According to social constructivist theory, people learn in a process of knowledge construction (Reigeluth, 1999). Knowledge is socially constructed by exchanging ideas, negotiating meaning, and sharing objectives and dialogue (Pea, 1993). Accordingly, the learning environments should provide activities and opportunities for students to articulate and reflect on material, and negotiate meaning with other students (Pena-Shaffa & Nicholls, 2004).

Online asynchronous discussion forums provide students with an appropriate context for meaningful learning. In such forums, students can collaborate, share and negotiate knowledge without the need to meet physically in person, or to work simultaneously. Therefore, these forums allow instructors and students to interact after class (Rossman, 1999).

However, most students do not post or reply to questions in discussion forums. A discussion forum fails if few or no messages are posted to it (Preece, Nonnecke, & Andrews, 2004). Therefore, teachers always have to act as facilitators to encourage students to discuss and guide them to produce beneficial dialogue in discussion forums (Mazzolini & Maddison, 2003). However, to create a successful forum, teachers have to spend considerable time and effort. One method for decreasing the load on teachers is to promote learners helping each other. Studies have developed several systems for locating the help of capable peers, such as Answer Garden 2 (Ackerman & McDonald, 1996), Intranet Peer Help Desk (Greer et al., 1998), and Expert Finder (Vivacqua & Lieberman, 2000).
differences between our system and these systems are both that the peer recommendation system not only finds capable peers to answer posted questions, but to motivate students to participate in discussion and that the peer recommendation system incorporates peer recommendations in the context of coursework activity.

The peer recommendation system selects two capable classroom peers to resolve a proposed question together with the questioner on a discussion forum. The reason for selecting two peers for each question is that we hope each question can be replied to by at least one student. If there is a question that a recommended peer can not answer or forgets to answer, then there is still the other peer to answer. To motivate those two peers to participate in the discussion, this system contains two assumptions: the students are interested in answering a posted question that relates to a programming exercise on which they are working, and students answer questions if they are requested. Based on the assumptions and students’ learning status, two capable peers for a posted question are selected in the following order: (1) students who receive and have previously submitted the same programming exercise as the questioner; (2) students who receive the same programming exercise as the one in which the question is raised, and who are also proficient in the main concepts required to complete the exercise; (3) students who have received the same programming exercise as the one in which the question is raised, and (4) students who are proficient in the main concepts required to complete the programming exercise in which the question is raised. This system can select two capable peers according to these rules. After selecting two peers to answer a question, the system sends an e-mail to these peers asking them to answer the question. The e-mail includes a hyperlink to the discussion thread, and some statements telling these peers that their comments are welcome.

Experiment

An experiment was conducted to evaluate the WCSS. The experiment was performed at the National Central University in Taiwan, and the subjects were 50 first-year undergraduate students. These students were all computer science majoring, enrolled in a mandatory course entitled “Basic Computer Concepts”, in which they studied Java programming (including structured programming, object-oriented programming, and graphical user interfaces).

The students learned in a blended learning environment. That is a teacher lectured for three hours weekly in the classroom and the students also learned via the web-based learning system after class. Seven coursework assignments were set over a 14-week period. Students were allocated two weeks to complete each assignment. During the first three coursework assignments (1-3), the students learned in the web-based learning system without the WCSS support. And each assignment consisted of three programming exercises, ranked as low, middle and high levels of difficulty, which were assigned to all students. The teacher determines the difficulty level of an exercise according to three rules. (1) an exercise that requires students to apply a programming concept with a complexity level of algorithm analysis that the teacher can design within one minute is ranked as low level of difficulty; (2) an exercise that requires students to synthesize two programming concepts with a complexity level of algorithm analysis that the teacher can design within five minute or an exercise that requires students to apply a programming concept with a complexity level of algorithm analysis that the teacher has to spend more than one minute on to design is ranked as middle level of difficulty; and (3) an exercise that requires students to synthesize more than two programming concepts or an exercise that requires students to synthesize more than one programming concept with a complexity level of algorithm analysis that the teacher has to spend more than five minutes on to design. During assignment 4, the students were able to use the e-dictionary and the peer recommendation system to solve their exercise problems. That means the students learned in the web-based learning system with the e-dictionary and the peer recommendation system but without PADS. During the last three coursework assignments (5–7), the students learned in the web-based learning system with the WCSS support. And three programming exercises selected by PADS were assigned for each student in each coursework assignment.

The proposed WCSS was evaluated using the following quantitative and qualitative data: (1) a set of web logs, which recorded all student interactions with the web-based learning system; (2) a questionnaire, which students completed at the course end; (3) notes from semi-structured interviews. These data were collected to answer the following questions:

- Whether the WCSS can increase likelihood that students do coursework within their ZPD,
- Whether the support provided by WCSS can help students complete their programming exercises,
- Whether the e-dictionary system is an effective tool in helping students complete programming exercises,
• Whether the peers selected by the peer recommendation system were willing and able to answer the assigned questions.

Results

Experimental results about the WCSS

After submitting an assigned programming exercise, the students were required to answer a question, “How do you complete the exercise?” Figure 7 presents the percentages of each answer in each assignment. The sum of the percentages of answers 2 and 3 in each assignment from assignments 1–3 (without WCSS) were obviously lower than the sum of the percentages of answers 2 and 3 from assignments 5–7 (with WCSS). This result may reveal that students with the support of the WCSS have higher likelihood of doing their coursework within their ZPD than without the support of WCSS.

![Figure 7. Percentage of each answer of the question “How do you complete the programming exercise”](image)

In order to investigate whether the result was statistically significant, a paired $t$-test was conducted. Because in the assignment 4, the students were able to use the e-dictionary and the peer recommendation system, but the three exercises were not selected by PADS. Therefore, the paired $t$-test compares the average number of answers 2 and 3 in an assignment during assignment 1-3 (without WCSS) with the average number of answers 2 and 3 in an assignment during assignment 5-7 (with WCSS). As shown in Table 4, the result was significant ($t$=-11.366, $p=0.00<0.05$), revealing that there were more exercises (means=1.81) that were completed with assistance in an assignment when using the web-based learning system with WCSS than without WCSS (mean=0.81). This result indicates that students with the support of the WCSS have higher likelihood of doing their coursework within their ZPD than without the support of WCSS.

To investigate whether WCSS can support the students in completing their exercises, a paired $t$-test was conducted by comparing the average number of answer 4 in an assignment during assignments 1-3 (without WCSS) with the average number of answer 4 in an assignment during assignment 5-7 (with WCSS). As shown in Table 4, the result was significant ($t$=9.376, $p=0.00<0.05$), revealing that there were fewer exercises (means=0.55) that the students could not complete before the deadline of an assignment when using the web-based learning system with WCSS than without WCSS (mean=1.59). In other words, there were more exercises that the students could complete before the deadline of an assignment when using the web-based learning system with WCSS than without WCSS.

There might be two reasons to explain why more exercises could be completed with the support of WCSS. One is that the e-dictionary and peer recommendation systems can provide effective support for students, so students can complete more difficult exercises. The other is that the PADS assigned easier exercises than the conventional method where the teacher assigns three programming exercises to all students. Because the average number of answer
(students can complete without any assistance) in an assignment during assignment 5-7 (mean=0.64) is approximately the same with the average number of answer 1 in an assignment during assignments 1-3 (mean=0.60), so the PADS in some degree did not assign easier exercises than the conventional method. Therefore, we tend to believe that the e-dictionary and peer recommendation systems could support the students in completing their exercises.

<table>
<thead>
<tr>
<th>Table 4. Comparing students’ responses when doing coursework without WCSS and with WCSS</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Without WCSS</strong></td>
</tr>
<tr>
<td><strong>Mean</strong></td>
</tr>
<tr>
<td>Answer 1 (completed without any assistance)</td>
</tr>
<tr>
<td>Answer 2 and 3 (completed with assistance)</td>
</tr>
<tr>
<td>Answer 4 (far beyond my ability)</td>
</tr>
</tbody>
</table>

* p-value < .05  ** p-value < .01

Experimental results about the e-dictionary system

The e-dictionary system was evaluated according to students’ perceptions. In general, an information system is more highly used and accepted when users have a more positive attitude toward it. Thus, a questionnaire, used in Davis’s technology acceptance model, was chosen as an evaluation tool to measure the effects (Davis, 1989). The questionnaire using 7-point Likert scales (from “1” which means “strongly disagree” to “7,” “strongly agree”) was devised with twelve items to investigate whether the students perceived that the e-dictionary system was useful and easy to use for completing their programming exercises. The Cronbach alpha reliability for perceived usefulness is 0.94, and for perceived ease of use is 0.94. The total Cronbach alpha reliability is 0.97; this implies the questionnaire data have a high reliability. Analytical results indicated that students’ attitude of perceived usefulness (mean=5.10, SD=1.35) and perceived ease of use (mean=5.20, SD=1.13) were all high, and thus the students perceived that the e-dictionary system was useful and easy to use for completing their programming exercises.

Experimental results about peer recommendation system

Table 5 lists the statistics of the students’ discussions in the discussion forum for each coursework assignment. The first column, “Question,” represents the number of questions posted on the discussion forum. The second column, “Reply,” represents the number of replies for each posted question. The third column, “Question solved,” represents the number of posted questions that were solved as determined by the teacher.

The peer recommendation system was able to be used after coursework assignment three was completed. This system selected two students for each posted question. Accordingly, the reply rate (Reply /Question) increased once the peer recommendation system was employed. The average reply rate from assignments 1–3 (without the peer recommendation system) was 0.43, which was lower than the average reply rate from assignments 4–7 (with the peer recommendation system), which was 1.14. The analysis may indicate that the peer recommendation system can motivate students to participate in discussion. The number of questions posted and answered declined in assignment 7, since the final exam was taken during this time, causing the students to spend less time on the web-based learning system.

Additionally, the questions-solved rate (Questions solved / Question) increased after assignment three. The average questions-solved rate (0.82) in assignments 4–7 (with the peer recommendation system) was higher than the average questions-solved rate (0.21) from assignments 1–3 (without the peer recommendation system). The peers selected by the peer recommendation system successfully solved 82% of the posted questions. These results may represent that most of the selected peers are willing to answer the assigned questions and are able to help questioners solve the questions. However, there were still 18% of the posted questions that can not be solved by the selected peers and the reply rate for each coursework session during assignment 4-7 was not anticipated. Theoretically, if the peer recommendation system requested two students to answer a question, then each question would be answered by at least two students. The reasons for this unanticipated reply rate and unsolved question rate may be: (1) some questions were too difficult for the selected students to answer, and (2) one student answered the question and, consequently, the second student did not feel obliged to answer the question.
Table 5. The statistics of the students’ discussions related to coursework assignments

<table>
<thead>
<tr>
<th>Assignment</th>
<th>Question</th>
<th>Reply</th>
<th>Reply / Question</th>
<th>Question solved</th>
<th>Question solved / Question</th>
</tr>
</thead>
<tbody>
<tr>
<td>Assignment1</td>
<td>4</td>
<td>1</td>
<td>0.25</td>
<td>1</td>
<td>0.25</td>
</tr>
<tr>
<td>Assignment2</td>
<td>0</td>
<td>0</td>
<td>0.00</td>
<td>0</td>
<td>0.00</td>
</tr>
<tr>
<td>Assignment3</td>
<td>10</td>
<td>5</td>
<td>0.50</td>
<td>2</td>
<td>0.40</td>
</tr>
<tr>
<td>Assignment4</td>
<td>27</td>
<td>25</td>
<td>0.93</td>
<td>20</td>
<td>0.74</td>
</tr>
<tr>
<td>Assignment5</td>
<td>27</td>
<td>39</td>
<td>1.44</td>
<td>25</td>
<td>0.93</td>
</tr>
<tr>
<td>Assignment6</td>
<td>17</td>
<td>19</td>
<td>1.12</td>
<td>15</td>
<td>0.88</td>
</tr>
<tr>
<td>Assignment7</td>
<td>8</td>
<td>7</td>
<td>0.88</td>
<td>5</td>
<td>0.63</td>
</tr>
</tbody>
</table>

Comments from interview with the students

Three students, SA, SB, and SC, were chosen randomly from three grade-based groups for in-depth interviews. The teacher interviewed with the three students after the course end. Some of the students’ comments that are useful for improving WCSS are summarized as follows:

- The suggested concepts listed on an exercise web page can hint which concepts I can use in completing the exercise. If I don’t know how to complete it, the hint can help me. But sometimes it may limit my creativity in using other concepts to complete the exercise. (responded by SB)
- Since each student received different programming exercises, it was not easy to find someone who received the same exercises as me. It forced me to complete the assigned exercises by myself. (responded by SC)
- Most of the assigned exercises are very difficult. I needed to spend considerable time completing each assigned exercise. I feel that it is not fair to give me difficult exercises. (responded by SA)
- The e-dictionary system was useful in learning programming concepts since it provided related learning materials, such as explanations, discussion threads, and programming examples. However, it did not help me analyze the algorithm of the assigned programming exercises. (responded by SB and SC)
- If I did not know which concepts I needed to complete a programming exercise, then the e-dictionary system was not useful in helping me complete the programming exercises. (responded by SB)
- The learning materials in the e-dictionary system were too simple and did not cover enough materials. It can not advance my skills. I prefer finding learning materials on the Internet for solving problems and learning. (responded by SA)

Discussions

The designed WCSS system, including PADS, e-dictionary, and peer recommendation system, provide three ways of assistance to support students in performing coursework within their ZPD. A result is statistically significant revealing that students with the support of the WCSS in a coursework activity have higher likelihood of doing their coursework within their ZPD than without the support of WCSS. In addition, the number of exercises that can be completed before deadline by students with the support of WCSS is significantly more than those done by students without the support of WCSS. It partly prove that the WCSS can provide effective support for students to complete their exercises. According to these results, we believe that WCSS can improve learning in the coursework activity by providing challenge and assistance.

The exercise web page that presents hyperlinks to the e-dictionary and peer recommendation system reminds students which ways of assistance can be used for completing their exercises. Moreover, the programming concepts listed in the concept section suggest which concepts can be used for completing the exercise. The WCSS providing explicit guidance and task-related assistance in the exercise web page can direct students to complete their exercises. It also reduce students’ cognitive load in help seeking activities (Aleven, Stahl, Schworm, Fischer, & Wallace, 2003). However, the explicit guidance may be ineffective where self-regulation is needed (Hill & Hannafin, 2001) and sometimes may hinder students from using different programming concepts for solving problems (SB responded this). The explicit guidance has both positive and negative effects for learning. Therefore instructional designers should consider this factor to provide learners appropriate support.
The PADS sometimes assigned very difficult exercises for advanced students (SA responded this). These exercises required more time to complete than the simple exercises. Hence, the advanced students needed to spend more time completing the assigned exercises than other students. They thought it was not fair. Although we did not focus on the assessment mechanism in this study, it remains an important issue to have a fair assessment mechanism when each student is assigned with the exercises of different levels of difficulty. This issue is also found in the researches related to CAT (Lilley, Barker, & Britton, 2004). In addition, to reduce the advanced students’ workload, the rules for assigning programming exercises may use the decision model and also consider students’ workload. Thus, if a student received very difficult exercises, then PADS can automatically reduce the number of assigned exercises from 3 to 2 or 1.

E-dictionary integrates distributed learning materials in a concept map structure. Hence, students can easily find the learning materials related to a programming concept and learn the concept from its explanation and related examples. The experimental results showed that the students highly accepted and used the e-dictionary. It may represent that the e-dictionary system is an effective tool in helping students complete programming exercises. However, some students complained that the e-dictionary system contained too little material and that some of it was too vague. Most of the learning materials within the e-dictionary were gathered from the web-based learning system and were categorized into explanation, example, and discussion. Because of the limitation of sources and categories of the learning materials, the e-dictionary may be more useful in helping students perform low-level cognitive processes, such as remembering, understanding, and applying programming concepts, than in helping students perform high-level cognitive processes, such as analyzing, evaluating, and creating (Anderson & Krathwohl, 2001). Future work can redesign the e-dictionary system so that students can collectively insert and edit the content in the e-dictionary system by themselves. The editing process is similar to that used in a wiki system, such as wikipedia (http://www.wikipedia.org/). Moreover, the e-dictionary also can provide personal note-taking tools that can associate collective knowledge representation with personal knowledge representation (Chen & Hung, 2002).

The reply and questions-solved rates were increased after the students were able to use the peer recommendation system. These results may represent that most of the selected peers are willing to answer the assigned questions and are able to help questioners solve the questions. A student who is notified to answer a question may recognize that he/she has a responsibility to answer it. Thus, the reply and questions-solved rate were increased. The students who just passively receive answers will not learn better than the students who actively participate in discussion. Therefore, peer recommendation can be thought of as a strategy for motivating students to participate in discussion. However, the number of questions posted was very few. The reason may be that we did not encourage students to post their questions in the discussion board in the beginning of the course (Preece, Nonnecke, & Andrews, 2004).

Limitations

In the proposed WCSS, the teacher and TAs were required to perform certain tasks such as identifying the concepts needed in a programming exercise, measuring the complexity level of algorithm in each programming exercise, constructing a concept map for this course, and extracting concept explanations from textbooks. These tasks require a significant amount of time. Therefore, nine TAs were employed to help the course teacher complete the tasks. Although these tasks required a significant time commitment, most were performed once, including constructing a concept map and extracting concept explanations from the textbook, and some tasks were completed by students, such as identifying the concepts used in a submitted assignment. Only two tasks, entering the examination grade and the coursework grade into the database, were repeatedly performed by the teachers and TAs. Therefore, the workload of teachers and TAs was only minimally increased.

The nine experts sometimes had some difficulty in consistently identifying the concepts required for completing an exercise. That is students can adopt different programming concepts to complete an exercise. For example, when solving the programming problem “Sum the integers between 1 and 100”, a student may use “for”, “while” or “recursive method” to control the loop. Thus, some experts may choose “for” and others choose “recursive method” as the main concept. The learning objective of the course was to train students in understanding and applying the basic programming concepts, but not in learning complex programming skills, such as algorithm analysis or data structure. Hence, the concepts used in each exercise must be clearly identified. For example, the above problem can be stated as “Sum the integers between 1 and 99, using recursive method.”
Conclusion

To increase the likelihood that students do their coursework within their ZPD, this paper has presented a Web Coursework Support System based on the notion that teaching and learning occurs only when assistance is provided to learners in their ZPD. Therefore, to assist students in performing coursework within their ZPD, WCSS provides three subsystems: PADS, which applies decision tree analysis methodology to select programming exercises with a level of difficulty suited to each student; an e-dictionary system that allow students to access distributed learning products related to a programming concept from a single web page; and a peer recommendation system that selects two capable classroom peers and motivates them to answer assigned questions. Experimental results show that the exercises-completed rate and the rate of the assigned exercises completed with assistance were increased. The findings indicate that WCSS can increase the likelihood that students do coursework within their ZPD by providing challenge and assistance. In addition, some suggestions for improving the coursework activity with WCSS support are proposed. Those suggestions consist of (1) considering the positive and negative effects of explicit guidance for providing learners appropriate coursework support, (2) considering the fairness of assessment when each student is assigned with the exercises of different levels of difficulty, (3) providing support for students to collectively construct knowledge in the e-dictionary system, (4) providing personal note-tasking tools for associating learning materials within the e-dictionary system with personal knowledge structure, and (5) conducting peer recommendation in the beginning of a course to promote interaction in the discussion forum.

References


