An e-Learning System for Programming Languages

Lih-Shyang CHEN[a], Emily YANG[b*], Shu-Han CHANG[a] & Young-Jinn LAY [c]
[a] *Department of Electrical Engineering, National Cheng Kung University, Taiwan*
[b] *Department of Leisure, recreation and Tourism Management, Southern Taiwan University of Science and Technology, Taiwan*
[c] *Department of Electrical Engineering and Engery Technology , Chung Chou University of Science and Technology, Taiwan*
*ccyang@mail.stust.edu.tw

Abstract: In this paper, we describe an e-Learning system that can help novices to learn how to develop some workable programs in a short period of time and assess their performance. The system takes advantage of computer and network technologies and combines the concept of flipped classroom to help the instructor and students in their teaching and learning activities. The preliminary study shows that the platform can indeed assess the students' performance and consequently help the students to learn programming languages more effectively and efficiently.

Keywords: programming language, e-Learning system, flipped classroom, student-centered learning

1. Introduction

Recently, the new concepts of the flipped classroom, student-centered learning (Jones, 2007), (Hannafin and Hannafin, 2010), (Johnson, 2013), (Crumly, 2014), (Young and Paterson, 2007), and problem-based learning in combination of advances in computer technology have led to a renewed interest in developing e-Learning systems (Richey, 2008), (Garrison and Anderson, 2003), (Nikhilesh and Karforma, 2012). In an extreme case, student-centered learning requires students to set their own goals for learning, and determine resources and activities that will help them meet those goals (Jonassen, 2000). However, in practice, this idea case may not happen very often unless the students are highly motivated by themselves. Therefore, in most practical cases, an instructor may use various blended learning methods for his pedagogical strategy. In particular, with the help of computer and network technologies, it is possible to develop an e-Learning system that can easily incorporate the concepts described above to form an interesting learning environment. In this paper, we develop such a system to effectively and efficiently help college students to learn how to write programs in an "introduction to computers" course.

In section 2, we briefly review the related concepts used in our system, such as flipped classroom, student-centered learning, and problem-based learning (Hmelo-Silver and Cindy, 2004), (Schmidt, Henk, Rotgans, Jerome, Yew, Elaine, 2011), (Neville and Alan, 2009). In section 3, we discuss the ideas of designing the e-Learning system. In section 4, we show how the system is used in practice, assess students' performance, and how the system can effectively and efficiently help the students to learn a programming language. In section 5, we describe the results of our experiments of the system. In section 6, we discuss the advantages and disadvantages of the system. Finally, we give conclusions in section 7.

2. Related Background

As far as an instructor is concerned, it is much easier to simply present whatever materials are in the textbook than to do something else such as diagnosing students' learning problems and making efforts

to help each individual student. Therefore, traditionally, most instructors prefer to give lectures rather than any other teaching activities. Consequently, in the traditional model of classroom instruction, the teacher typically gives lectures, is the central focus of a lesson, and is fully in control during the class time. Since the instructors simply disseminate the knowledge in the textbook, normally students do not really find too much difficulty to learn the pure knowledge. In particular, in most Asian countries normally students just passively listen to the lectures and keep quite instead of actively asking questions in classroom even though they do have some problems with the materials the instructor teaches. In other words, most classroom are didactic and entirely content oriented without considering whether the students really learn what they are supposed to learn. In order to ensure that the students really learn the teaching materials, instructors normally assign some homework for the students to take home and exercise the related works. Generally speaking, students typically do not really find any problems with the learning materials until they are asked to apply whatever they learn to solve a real problem. In other words, they usually find difficulty when they do their homework that are assigned to them to do after class.

The concept of flipped classroom is an attempt to remedy this problem. It is an instructional strategy of blended learning that reverses the traditional class arrangement by delivering the learning contents outside the classroom, often on-line, and moving the activities such as doing homework and discussing specific problems in the classroom. In this case, whenever they encounter any problems, they can simply ask the instructor or classmates right away and are able to learn more than that in the traditional class. This arrangement also fits itself into the concepts of problem-based learning and student-centered learning.

However, in this case, the instructor may still face the problem that students may be doing something else instead of really working hard on their assignments. In particular, when there are many students in the classroom, it is virtually impossible for the instructor to assess every student's work simultaneously. As a result, many students may be working on something else that may be more interesting to them, such as a game. Eventually, the students' performance may not be as good as what we originally expected.

Therefore, we set out for developing an e-Learning system that can easily assess every student's performance implicitly and explicitly so that the instructor can always know the learning status of every student. Consequently, the instructor can do whatever is necessary to help the students in learning activities.

3. The Ideas of Designing the e-Learning System

Generally speaking, a student in an information engineering department of a university is supposed to take a programming language course in which he should learn how to write a workable program. However, currently in Taiwan, many students still can not develop a workable program after taking such a course. In fact, this is one of difficult problems we are facing in most universities. Based on our experiences, the major reason that causes this problem is that students who fail to do so do not really try hard enough to write a program and test it by themselves. Instead, whenever they encounter some problems, they simply quit and simply plagiarize other classmates' work with minor modification so that their programs do not look exactly the same as others. In some cases, the students may indeed try to write a program. However, they encounter a lot of problems and nobody can really sit next to them to help them to solve the problems. In this case, the plagiarism seems to be unavoidable. Since it is difficult and time consuming for the instructors or teaching assistants to really check through all the students' homework to find whether there is any plagiarism, those students can usually get away with it. Consequently, those students still can not write a workable program after finishing the course. We believed that if we make use of computer and network technologies in combination with the concept of flipped classroom, we should be able to solve the problem to some extent.

The features that we want the system to have are the following: (1) We should try to somehow "force" the students to really write and debug a program on their own. (2) We should somehow assess whether they are really do the work. In other words, we should make use of computers to collect formative data for monitoring and assessing the students' performance as much as we can. (3) Since the students are novices, most of them have difficulty to discern what is a good/bad program, how to solve

a specific problem, and so on. Therefore, we should have a mechanism to show them some real examples done by some students and explain to them about some key points so that students can fully understand how to solve a real problem and how to avoid some mistakes. (4) We should try to make use of the concept of the flipped classroom which suggests that we should try to teach less in the classroom and ask students to do exercises and discuss various problems in the classroom. The students can study some more learning materials after class by themselves. In particular, as far as programming languages are concerned, there is no profound theory behind them. All the programming languages just have a lot of rules that can be easily understood if the instructor can emphasize the important concept in his lecture. The real difficulty of the programming work is the applications of those rules in a real case. In other words, we do need to use some real examples to explain to the students about the insight into the applications of programming languages. Therefore, what the students should do is to really exercise the programming work.

4. How the System is Used in Practice

At the beginning of each class, the instructor can click at the "上課" button and "出席狀況" button as shown in Figure 1.

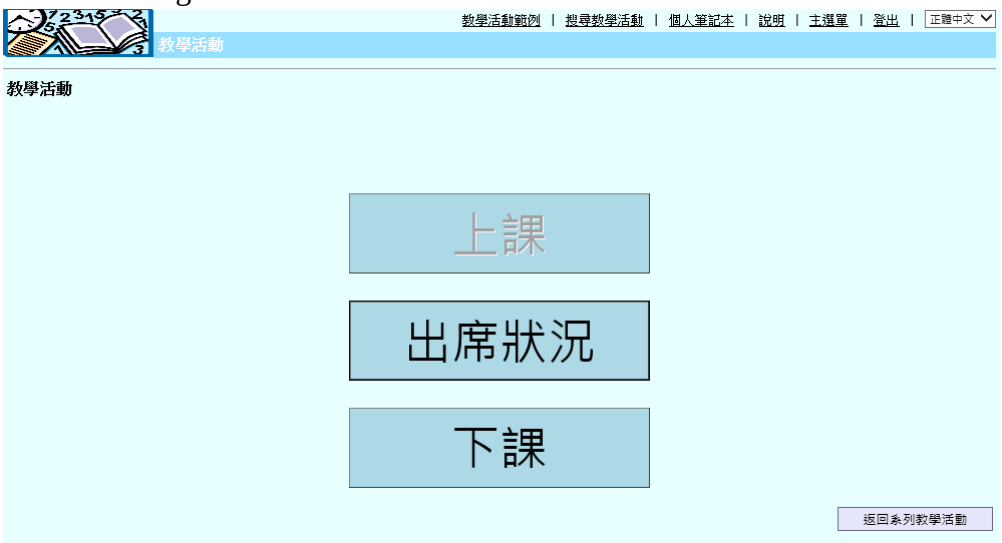

Figure 1. The webpage for starting out the class

The system will show all the students' names in the webpage as show in Figure 2. As time goes on and the students log in the system, the students' icons will be lit up to show that the students attend the class. In this case, the fourth student from the left has logged in. The instructor can check the attendance situation of the class.

Figure 2. The system will show the students' names in the webpage.

In each class, the instructor gives a short lecture to explain the essence of a program statement, such as a “for” loop. Then, a simple example is given to explain how the “for” loop really works in practice. At this point in time, the students should have a basic idea about a “for” loop without any difficulty in most cases. The instructor can begin to give a question and ask the students to write a program to solve the problem. For instance, the question can be “accept a natural number from the user and compute the sum of even numbers that are less than the given natural number”. The instructor can click at the button circled by the red rectangle in Figure 3.

Figure 3. The webpage for the instructor to give an assignment

The instructor can select whether he wants to give an oral question, an existing exercise question, or generate a new exercise question. For instance, he can give an existing exercise question in the exercise database. The exercise database is organized in a hierarchical structure as shown in Figure 4.

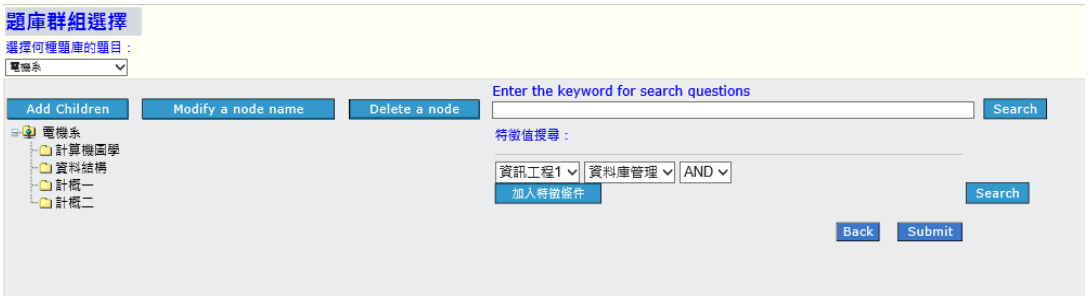

Figure 4. The existing exercise questions are organized in a hierarchical structure

The instructor can select which types of the questions he wants as shown in Figure 5.

Figure 5. The different types of the questions in the system.

The instructor can select one of them for this short quiz or exercise. The system will show the contents of this exercise for the instructor to confirm whether this is really what he wants. The instructor can also give an assignment on the spot if he wants. If the instructor confirm this exercise, the system will ask the instructor to set the time duration for this exercise as shown in Figure 6.

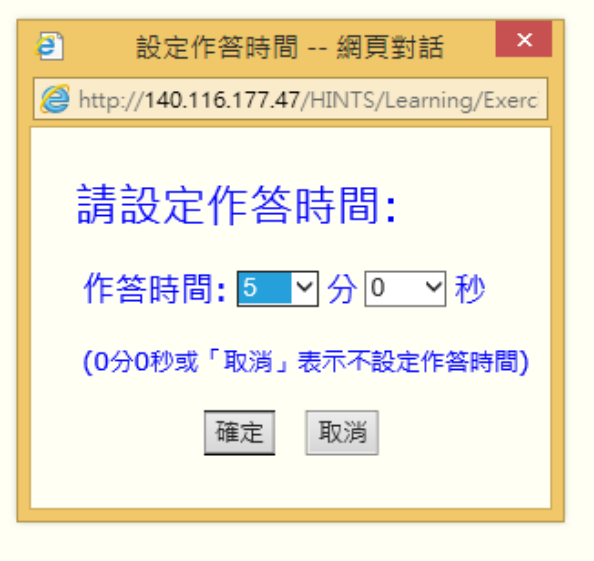

Figure 6. The instructor can set the time duration for the current exercise.

If the instructor gives two questions in a row, the students will see that the first two exercise questions are open as shown in Figure 7. The students can begin to work on their programs (the exercise) on their computers in the computer room in the department. During this time period, the instructor can go around the computer room, look for students that need help, and help them to solve the problems immediately. When the students finish up their programs, they should upload their programs by clicking at the button “”.

Figure 7. The first two questions are open and one student has submitted the first answer to the question one.

The system will show their submission status for each exercise so that the instructor can see how the students perform. If the time is almost up and most of the students are still working on the program, the instructor can extend the time period so that the students can continue to work on it. When the time is up, the system will show the exercise is closed and the students can not upload their programs. At this time, the instructor can further extend the time period if he wants by clicking a button. The instructor can also select whether the students can see each other's program or not. If they can, the students can click at a button to see other students' programs. If they can not and the instructor wants to show the students' programs, the instructor can select one student's program to show all the students and explain the good/bad points in the program to all the students. In Taiwan, all the computers in the computer room are equipped with a share mechanism for the instructor to show the contents of his screen to all the students by clicking at a physical button. Therefore, it is very easy for the instructor to make use of the students' program as a real example to explain mistakes that usually made by the students or virtues of good programs.

5. The Results of the Experiments of the System

Since the system will automatically compute the statistics of the time duration of the students spent on each exercise and the submission rates of each exercise, the instructor can easily understand how the students perform. The students are told that their performance will be assessed by the system automatically. Since the students are asked to work on their programs in the computer room, they are much more likely to really work on their programs in particularly while the instructor is going around the computer room. Furthermore, the students can easily get help from the instructor or other classmates whenever they need the help. The instructor can also check the students' programs at the same time and understand how the students perform.

At the very beginning of the class, when the students were asked to work on a program, the students were still not used to the new learning style and were not eagerly work on their programs. They were just hanging around there to wait and see what is going to happen. In other words, they still thought that the exercise was similar to other laboratory work in which students were normally chatting or joking around while doing some experiments.

After the first exercise, they realized that everything, such as whether they submitted the results, was recorded. They could see through the system who had submitted a program and who has not. Furthermore, the instructor could see the submitted programs and even showed someone's program anonymously to the whole class and made comments on it. As time went on, they realized that the system would record everything and the instructor could check who was not really working hard if he did not really submitted his programs. They gradually realized that they could no longer fool around just like what they did before. There is no kidding. The system was actually for real. Therefore, they were getting serious about each exercise and gradually became more and more active learners, and really tried hard to finish up their programs. As a result, the submission rate became higher and higher as time went on.

We could also easily observe this situation from the rates at which the students asked questions. As time went on, more and more students began to ask questions although traditionally Asian students tend to hesitate to ask instructors for the problems they encountered in their study. However, in our case, the students were under the pressure to finish up their exercises as soon as possible, they did break their silence and more vigorously asked questions in an attempt to solve their problems. Furthermore, in order to encourage the students to ask good questions, the instructor did give extra bonus points if a student asked an interesting or meaningful questions. This arrangement indeed worked very well since many good questions were asked in the classroom and many students' misconceptions were indeed found. In Figure 8, the instructor and students can check the datamining webpage to see all the data collected by the system. The instructor can select what kind of statistics he wants to see.

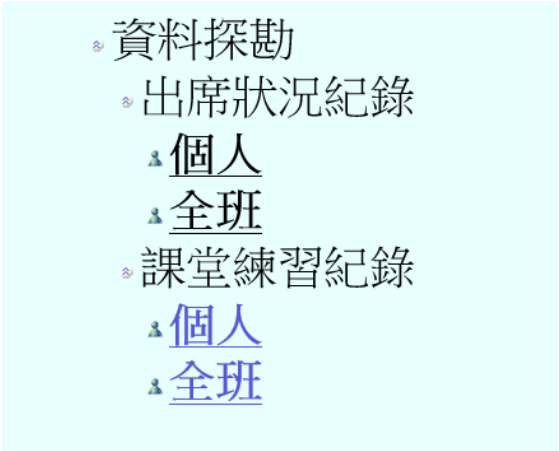

Figure 8. The datamining webpage of the system.

The instructor can check the attendance status for each class as shown in Figure 9.

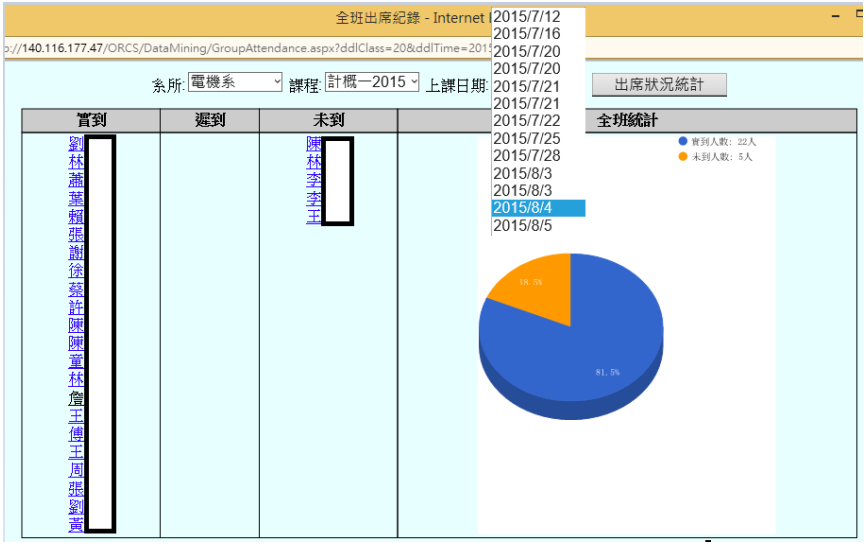

Figure 9. The system shows the attendance status of the class.

The instructor can also see the attendance status of every class as shown in Figure 10.

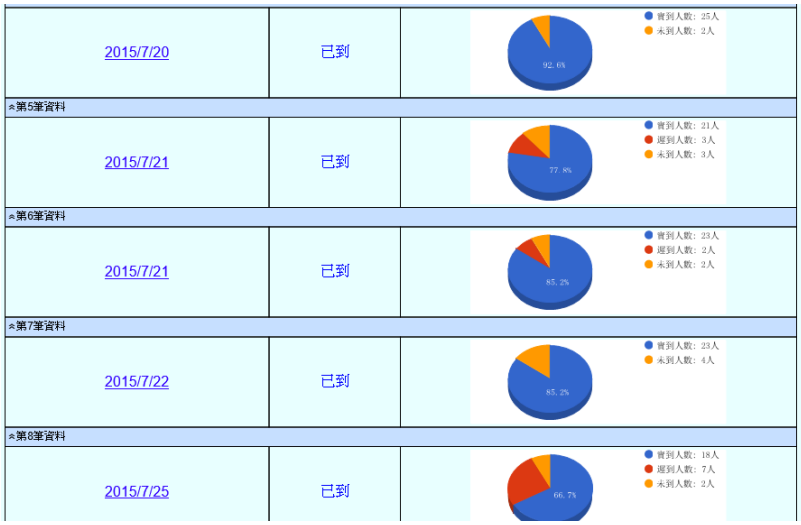

Figure 10. The attendance status of each class.

The attendance statistics of each student can also be seen in Figure 11.

使用者名稱	準時到課	遲到	未到	實到(%)
劉	8	0	5	8(61.5%)
林	8	0	5	8(61.5%)
陳	3	0	10	3(23.1%)
蕭	6	1	6	7(53.8%)
葉	8	2	3	10(76.9%)
賴	8	3	2	11(84.6%)
張	6	3	2	9(81.8%)
林	0	2	11	2(15.4%)
謝	7	0	6	7(53.8%)
徐	5	1	7	6(46.2%)
李	3	2	8	5(38.5%)
李	6	0	7	6(46.2%)
蔡	6	2	5	8(61.5%)
許	8	1	4	9(69.2%)
陳	7	0	6	7(53.8%)
陳	8	1	2	9(81.8%)

Figure 11. The attendance statistics of each student.

In Figure 12, the system shows the submission status of each class. We can see that at the very beginning, the submission rate is almost zero. As time went on, the submission rate did increase dramatically.

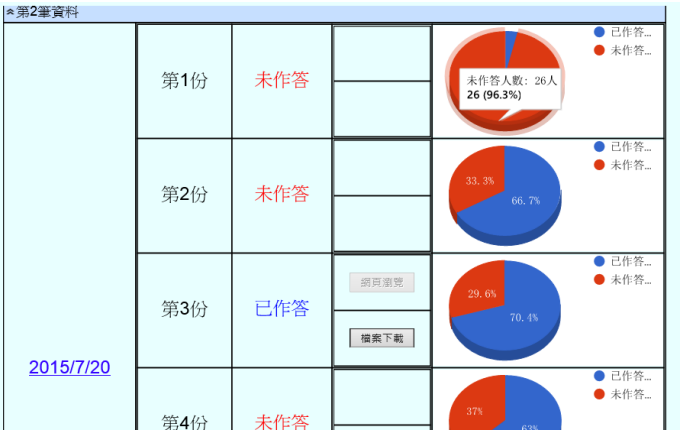

Figure 12. The submission rates of the exercises are displayed.

Since the system will automatically compute the statistics of the time duration of the students spent on each exercise and the submission rates of each exercise, the instructor could easily understand how the students perform. The students were told that their performance would be assessed by the system automatically to some extents. The students could easily get help from the instructor or other classmates whenever they needed the help. The instructor could also check the students' programs at the same time and understood how the students performed.

6. The Advantages and Disadvantages of the System.

6.1 The Advantages of the System

The students feel that although this learning style kind of "forces" them to do the programming work, they can learn the work more quickly and easily. The programming work is really not as hard as what they originally thought. Whenever they have problems in the programming work, they have opportunities to ask the instructor or classmates directly to solve the problems immediately instead of getting frustrated and giving up eventually. The formative data we collected during the class time and the statistics we computed show that the students indeed keep improving their programming skill as well as the motivation to learn. This formative assessment instead of a summative assessment can play an important role to monitor students' performance and really improve students' programming skills over time. As the students keep improving their programming ability, they are more interested in writing programs and have much strong sense of achievement whenever they make their programs workable. Consequently, they become more active learners and even try to develop programs with more features than what was given by the instructor.

6.2 The Disadvantages of the System

The downside of this learning style is that the instructor who make use of the system should (1) be enthusiastic about teaching and (2) have enough real experiences about the developments of software systems. For the first point, since the instructor should go around the classroom and look for the students' needs for help, this task is much harder than simply deliver the lecture of the learning materials that are in the textbook and he is familiar with. For the second point, since most professors focus on the theoretical aspects of teaching materials and may forget about all the detailed programming skills, it is hard for most professors to go back to the nitty-gritty details of the programming work that they did maybe more than ten years ago. This may pose a big challenge to a professor who is not really enthusiastic about the teaching and is normally evaluated about his research work instead of the teaching efforts by the institute that he is affiliated to. Furthermore, some students' programs may have some strange bugs. The professors may have to really spend time on the students' programs and have enough experiences to find bugs and solve the problems for the students.

7. Conclusions

From the data collected by the system, we can see that the students do become much more active learners. We also find that the students can really develop workable programs that they were not able to do in the past. The system can really assess the students' performance and help the students to learn programming languages effectively and efficiently. Students' programs or other example codes can be stored as learning scaffolding. Furthermore, the students' works can be stored as their portfolios that may be beneficial to improve students' learning achievement in the long run. The system, as it stands today, still has some problems that need to be corrected in the future. First of all, the system is not able to automatically check whether the students get right answers or not. This may not be a difficult work to do in most cases since we can automatically generate some random numbers as the input to a program that is a correct one and compute the correct answer. Then, we can compare the correct answers with the students' answers. Consequently, the system may be able to check the correctness of the studnets' programs to some extents. Secondly, based on the current technology, we are still not able to

automatically diagnose the students' programs if something does go wrong in their programs. We still need to rely on human efforts to do the diagnosis which may be a difficult and time consuming work. We believe that we might be able to solve the first problem in the near future and make the system even more useful in the future.

References

Jones, Leo. (2007). *The Student-Centered Classroom*. Cambridge University Press.

Hannafin, M. J., & Hannafin, K. M. (2010). Cognition and student-centered, web-based learning: Issues and implications for research and theory. In Learning and instruction in the digital age (pp. 11-23). Springer US.

Johnson, Eli (2013). The Student Centered Classroom: Vol 1: Social Studies and History. p. 19. ISBN 1317919491.

Crumly, Cari (2014). Pedagogies for Student-Centered Learning: Online and On-Ground. p. 26. ISBN 1451489536.

Young and Paterson, (2007). Teaching Nursing: Developing a Student-centered Learning Environment, p. 5. ISBN 078175772X

Jonassen, D.H. (2000). Revisiting activity theory as a framework for designing student-centered learning environments. *In D.H. Jonassen & S.M. Land (Eds.), Theoretical Foundations of Learning Environments (pp. 89–121)*. Mahwah, New Jersey: Lawrence Erlbaum Associates.

Richey, R.C. (2008). Reflections on the 2008 AECT Definitions of the Field. *TechTrends 52 (1): 24–25. doi:10.1007/s11528-008-0108-2.*

Garrison, D. Randy and Anderson, Terry (2003). Definitions and Terminology Committee. *E-Learning in the 21st Century: A Framework for Research and Practice*. Routledge. ISBN 0-415-26346-8.

Nikhilesh and Karforma (2012). RISKS AND REMEDIES IN E-LEARNING SYSTEM. International Journal of Network Security & Its Applications (IJNSA), Vol.4, No.1, January 2012

Hmelo-Silver, Cindy E. (2004). Problem-Based Learning: What and How Do Students Learn?. *Educational Psychology Review*, 16 (3): 235.

Schmidt, Henk G; Rotgans, Jerome I; Yew, Elaine HJ (2011). The process of problem-based learning: What works and why. *Medical Education*, 45 (8): 792–806.

Neville, Alan J. (2009). Problem-Based Learning and Medical Education Forty Years on. *Medical Principles and Practice*, 18 (1): 1–9.