Combining Virtual Learning Environment and Integrated Development Environment to Enhance e-Learning

Majlinda Fetaji 1, Suzana Loskovska 2, Bekim Fetaji 3, Mirlinda Ebibi 4
[1]*South East European University CST Faculty, 1200 Ilindenska bb, Tetovo, Macedonia*
[2]*St. Cyril and St. Methodius University, ETF Faculty, Karpos 2, Skopje, Macedonia*
[3]*South East European University CST Faculty, 1200 Ilindenska bb, Tetovo, Macedonia*
[4]*South East European University CST Faculty, 1200 Ilindenska bb, Tetovo, Macedonia*
Emails: [1]m.fetaji@seeu.edu.mk, [2]suze@etf.ukim.edu.mk , [3]b.fetaji@seeu.edu.mk
[4]m.ebibi@seeu.edu.mk

Abstract. *The research was undertaken having in consideration two hypothesis. The first hypothesis is that integration of virtual learning environment (VLE) and integrated developing environment (IDE) for programming in Java language will contribute in improving the efficiency and quality in learning because of the enhanced graphical user interface and the "hands on approach".*

The second hypothesis is that the designed graphical user interface of the virtual learning environment will contribute in facilitating its use by improving the results of the learning process, increasing the user-satisfaction and attention during learning that implicates improving the overal efficiency of learning programming in Java. The aim of this paper was to discuss difficulties and disadvantages of learning programming in traditional method, and to investigate for new e-learning strategies by combining e-learning and developing environment whereas the importance of the pedagogical approach is discussed and one is adopted in the design.

The usability of the created virtual environment was reviewed, in order to assess and propose solutions to the identified issues.

Keywords. Development environment, virtual environment, e-learning, enhanced learning

1. Introduction

During the last decades, due to the development of information and communication technology and the raising impact of the Internet, an access to a huge amount of information is enabled world wide. This offers new opportunities to acquire knowledge any time, anywhere regardless to the previous constraints, time and location. More and more information in a daily basis is presented in digital and multimodal form. In order to use all this information in the process of learning electronic environments are created and used. The impact of these technologies is reflected in the increased utilization of e-learning systems and virtual e-learning environments for learning. However there is a certain sceptisizm regarding e-learning and virtual environments efficiency lately. This is the reason why we have analysed e-learning systems and virtual e-learning environments. We are in an opinion that further research needs to be conducted to design a grounded theory that would focus on developing a good and efficient system for learning.

2. Motivation

Today, there are a number of e-learning systems and virtual environments for learning programming, but still there is not a clear how to design a good and efficient learning system and environment. There are many reasons to study this issue: 1. The student's need to have an interactive environment for learning and applying programming. 2. The complexity of the existing e- learning systems and environments that redirects the aim of learning into learning to use the environment. 3. To keep attention and motivation of the students to learn programming, the learning environment needs to have an efficient and attractive user interface. 4. Increasing the user interactivity to the learning environment that can stimulate more emotions that improves the quality of learning.

3. Disadvantages of the traditional methods of teaching and learning programming

Learning to program in object oriented language is difficult for novices in the traditional

classroom method. Research on teaching and learning computer programming indicate that novice programmers have several difficulties with computer programming concepts(as cited in) [6]. Programming is not a single skill but a multi-layered hierarchy of skills, many layers of which need to be active at the same time [7]. Programming demands a great deal of implicit knowledge which is difficult for lecturers to make explicit and which cannot easily be transmitted directly to students. In order to gain knowledge, students need to go beyond explicit information to construct experiential implicit knowledge [6]. According to [8] "programming knowledge cannot be directly transmitted from instructors to students to construct - it must be acquired actively by students" by practicing the acquired knowledge. Novices spend little time in planning and testing code. One of the problems might be considered the "long way to the program outcome", that requires a number of steps to be accomplished that are often disadvantageous. The traditional method of listening lecturers and reading additional materials are elementary in constructing knowledge. But the teacher must transmit new ideas of programming concepts and it is very difficult to concretize abstract concepts of programming with the concept "on the table". Most importantly, repetition by practicing will ensure that the knowledge is retained. So students need to do a lot of practice. Unfortunately, traditionally in classroom, they encounter many difficulties in the "the editing, compiling, and testing cycle" while creating an executive object was impossible with the concept ex-cathedra. Therefore, a simple text editor was used to write the code. To compile and execute the Java code, the Sun Java compiler - Javac was invoked writing a strict syntax-ed command that initiated an increased rate of errors.

There are a number of other disadvantages of teaching and learning in the traditional method in a classroom:

1. The traditional method of learning is instructor-centered and depends on the methods the instructor uses. 2. The method of teaching lectures on the table, even the visual format of the lectures in a computer, is not sufficient. 3. To overwhelm all the elements of the process of learning to program, each element must be practically tested. 4. The traditional method is limited in time, place and time duration of the class. 5. The wide range of previous experience and background knowledge from novices to advanced programmers and the difference in their capabilities might be disadvantageous to follow the lecture stream on the table (some might need to do revision on the lectures that is impossible in the traditional way of learning). 6. Using the "hands on approach" that means the learned concepts can be tested and applied immediately, also is impossible in the traditional way of learning. 7. In this method of learning, the students are more passive. Through structuring the learning content in which problems are presented and enabling "revision on the lectures" any time/ any where through an e-learning environment, it is offered the opportunity to be active in the learning process that encourages a more natural style of learning.

All these arguments can contribute in loosing the interest for learning the programming language, in decreasing the satisfaction and reliability while learning, in emerging difficulties while learning and these will results in dropping-down the course for learning the programming language. To outrun these difficulties and to complement the demerits of the traditional method of learning to program, today there are a number of virtual learning environments, developing environments and software programs and applications to help students in learning to program. There are new strategies developed that outweigh the traditional framework for learning programming. The existing virtual environments and the usage of computers in learning and practicing programming can enhance the integration of theory and practice of programming that further more promotes new strategy of teaching and learning programming whereas immediately what have been learned can be tested instead of memorizing theoretically transmitted programming concepts. The most existing VLE for learning programming offer just learning environments, even more those that offer developing environment allow practicing just in certain time of learning session, thus our virtual learning environment offers an integrated learning and developing environment flexible in aspect of pacing the learning/testing process.

Besides, many researches have been done in a context to find the best method to teach and learn programming and to improve the acquisition of programming skills. Generally, as cited in [6], one of the main factors that influence the process of the development of programming skills and skills for solving problems using programming is the learning environment. Due to our experiences and that of the other colleges, a conclusion is

drawn that teaching object oriented programming is much easer if an electronic environment is used. The new learning methods that are developed are e-learning systems and virtual learning environments that tend to gradually substitute the traditional methods of learning in classrooms and the classroom learning environment are complemented with electronic learning environment.

4. New strategies for teaching and learning programming - Virtual learning environments

Over the last years, the education and learning and teaching have been influenced by the rapid technology development. That is the learning process has been changed towards more interactive learning activities and authentic experiences according to [1]. The new learning environments are technology enhanced and supported and computer based environments called virtual learning environments. [2]defines virtual learning environments as computer-based environments that are relatively open systems, allowing interactions with other participants and access to a wide range of resources. Such environments foster the "any time/any place" learning model that is not only a different way of delivering knowledge, but also a powerful means of creating knowledge. These new ways potentially have a wide range of advantages over traditional environments (e.g., convenience, flexibility, lower costs, currency of material, increased retention, and transcending geographical barriers) according to [3]. To help novices to learn programming we have focused our research on developing a virtual environment to facilitate learning to program in a sense of offering an electronic environment that should meet all the users needs and overrun the demerits of the traditional method of learning. Usually while developing virtual environments pedagogical aspect is left behind without consideration. Therefore, to develop a quality e-learning virtual environment for learning Java we have focused on the pedagogical concept of the e-learning solution.

5. Pedagogical background of an e-learning virtual environment

A research is made on how to design a quality e-learning. According to [4] to design our e-learning solution we have followed the approach that design and use of e-learning must be grounded in a learning theory approach. "In order to develop the use of e-learning from a pedagogical point of view, it is therefore not enough to study the existing practice. Instead, it is necessary to have an understanding of theoretical principles of the learning process and of the ideal learning environment" [5]. The learning environment is important because it models the learning process of particular course in a technological medium, so we have to ideally model the learning process.
This means that the design of e-learning can not be based only in the existing practice, it is necessary to understand the relation between theory and practice to ensure that the design of practice is founded on the learning theory. This concept is shown in the following figure:

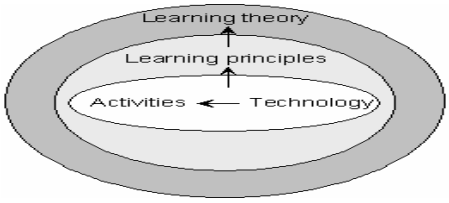

Figure 1. - Theoretically grounded evaluation of technology [2]

We have followed this concept as a pedagogical background of our e-learning solution. It describes that the different learning activities that are driven in the learning environment are supported by the e-learning technologies stated above. The learning principles are formed by the learning activities to be done to produce the learning outcome. The learning activities are crucial to define the features and abilities the learning environment has to support to facilitate learning and thus supported by the technology. According to the concept of grounded design in [4] that is defined as the systematic implementation of processes and procedures that are rooted in established theory and research in human learning", the implementation of the learning activities are rooted in a learning theory and how novices learn programming.

6. Cognitive and Constructive issues in modeling virtual environment for learning programming in Java

The cognitive and intellectual abilities of learners are crucial in the process of learning to

program in an object-oriented programming language. Learning to program in an object-oriented programming language is a complex process where the cognitive learning from lectures approach is not self sufficient. The research literature shows that programming involves a high level of cognitive development and "incorporates cognitive science studies of mental activities that are involved in programming" [11]. Computer programming is highly complex since "it involves subtasks that draw on different knowledge domains and a variety of cognitive processes [17]. It involves a number of abilities that interrelate with the learner's background knowledge and experience, reasoning skills, processing skills, a wide range of comprehension strategies, and abilities for problem-solving and continual use of problem solving and design skills. Therefore, there is variety of cognitive activities involved in computer programming where "cognitive skills or cognitive mismatch also seem to play a significant role"[10]. To support the cognitive learning in the process of learning to program, a constructive approach of creating knowledge should be enabled. Programming cannot be learnt without doing a lot of practice. When learning to program, it is essential that students are given the opportunity to practice in an environment where they can receive constructive and corrective feedback [8]. The biggest problem of novice programmers is not understanding of basic concepts but learning to apply them. So, some basic test and debugging strategies should be taught [9]. These constructive activities should be made by the learner themselves in an active, motivating manner that take place in the learning environment. The combination of the two approaches would give a better result on the process of learning. The model of the developed learning environment is founded on the learning activities that depend on the cognitive and intellectual abilities of learners and their abilities to individually construct knowledge.

7. Pedagogical concept adopted in the design of the virtual environment for learning programming in Java

The pedagogical concept adopted in designing the e-learning virtual environment to learn Java is: 1) Our e-learning solution for learning programming language Java is grounded on the cognitivist and constructivist learning theory; 2) The learning environment is modeled by cognitive learning activities supported by the structured learning content into learning objects that are integrated as online help content; and constructive learning activities supported by the editing-developing environment which enables activities of constructing solutions to problems, in the sense of enabling "edit-compile-test cycle" to different programming concepts of given examples (in the help content) or examples created by the user by simple "clicking the appropriate icon" on the interface of the environment. 3) The independent student work supports their individual cognitive abilities to perceive the learning content the best way possible and process it into knowledge; and it also supports the individual and subjective construction of knowledge 4) The students work is based on their independent exploration of the learning content where they learn; and their independent construction of knowledge by testing what they have learned and creating new solutions of the given examples or new problems.

8. Advantages and disadvantages of learning programming using the designed virtual environment for learning Java

The virtual learning environment for learning to program in Java will have a simple GUI and will be easy to understand and use. To support the novice programmers our project provides a set of specially designed tools. It includes an editor for editing programs and file manipulation, visual tools for program construction that can be conducted entirely through menu interaction and help content modularized in reusable learning objects that are all presented within a single user-interface framework. The virtual environment offers a developing environment that enables the "hands on approach" that helps students to improve the quality in learning in a sense of immediate testing what they have learned while they construct knowledge by themselves instead of memorizing theoretical concepts. This will offer students to learn and practically test easily the programming concepts by themselves, to self-pace the process of learning, flexibility in time/place of learning, flexibility in accessing the content, multiple revisions to the content, with no limitation on duration of a learning session.

Of course, there are still some disadvantages of using the developed virtual environment against the traditional method of learning Java:
1) The acquisition of some skills and concepts of programming depend on direct face-to-face contact with the instructor. 2) The classrooms enable to get an instant feedback from the learners which are very important in the process of learning. 3) The students that can not learn without help are disadvantaged. 4) The face-to-face training with an instructor leads to greater interaction during learning where the learner may acquire knowledge from the instructor and that leads to greater success.

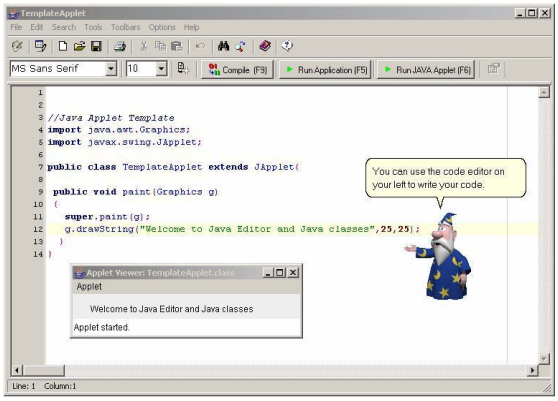

Figure 2. Virtual Environment for learning Java

9. Usability Testing

We conducted usability testing based on performance measurement to quantify usability requirements such as time to complete a task, time to learn, rate of errors and subjective satisfaction defined by task scenario using the traditional method environment and the developed integrated virtual environment Java. Also we made an evaluation by direct observation of users while they were performing different tasks by using the traditional method environment and the developed integrated virtual environment, and users from two different classes were observed. What do we evaluate in terms of usability: 1) functionalities: can the user perform the requested tasks?; 2) time: are the tasks performed in a reasonable time?; 3) satisfaction: is the user satisfied?; 4) Mistakes: does the user make a lot of mistakes? 5) comparison, in particular with text based interface tool.
TASK Scenario:
User: I want to create java application, compile it and execute it (run it).
Background: The editor (in the traditional method, an notepad editor is used) is empty has no text content, the application window has the focus.
Task: In traditional manner User starts the notepad editor and starts writing the code. The user must save the file as java source code with (.java) extension. To compile and run the code, the user must activate the command prompt window and write strict syntax-ed commands.

Using the virtual environment

User clicks the template button at the toolbar and clicks the compile button which prompts him to save the files as java source code with (.java) extension, and then the user clicks the run java application button. We can view the results of the usability testing in the two used environments in the following tables:

Table 1. Usability research for Class-1 in the Traditional method and the developed virtual environment

Usability Attribute	Measuring instrument	Value to be measured	Traditional method environment	Integrated virtual environment
Time to learn	Task Scenario	Time to complete task	600	400
Speed of performance	Task Scenario	Time to complete task	600	400
Rate of errors	Task Scenario	Number of errors	7	2
Subjective satisfaction	Task Scenario	Satisfaction degree of users	2	5
* number. Subject satisfaction scale: very high 5, high 4, average 3, low 2, very low 1				

Table 2. Usability research for Class-2 in the traditional method and the developed virtual environment

Usability Attribute	Measuring instrument	Value to be measured	Traditional method environment	Integrated virtual environment
Time to learn	Task Scenario	Time to complete task	500	300
Speed of performance	Task Scenario	Time to complete task	500	300
Rate of errors	Task Scenario	Number of errors	5	1
Subjective satisfaction	Task Scenario	Satisfaction degree of users	1	5
* number. Subject satisfaction scale: very high 5, high 4, average 3, low 2, very low 1				

10. Conclusion

The conclusions may be summarized as follows: The variables provide quantitative and objective and subjective data. The experiences introduced suggest the positive effects of using the Java Editor in classroom teaching/learning. The user interactions to the interface have shown high degree of user friendly concept embracement of the developed virtual environment.

Using this kind of user centered approach in building our graphical user interface and involving the users at each stage of the development and evaluation of the interface we have concluded that it resulted in a very user friendly graphical user interface and usable learning environment. According to the empirical research results the new developed graphical interface has several advantages: system is easier to use and has better performance rate than the textual command line based interfaces, it is less expensive and less time consuming, a greater accuracy in the process of writing the code has been achieved, achieving a task is much easer. Users are more confident and satisfied using the interface of the virtual environment. We recommend a pedagogical approach to designing virtual learning environment and particularly for learning programming the learning environment to be grounded in cognitive and constructive learning theory. Also, for usable learning environment we recommend a user-centered design where users are involved in the design life-cycle.

11. References

[1] Malone, P., Schryer, C. & Rossner-Merrill, V. (2000). Combining Instructional Models and Enabling Technologies to Embed Best Practices in Course Instructional Design. In P. Kommers & G. Richards (Eds.), Proceedings of World Conference on Educational Multimedia, Hypermedia and Telecommunications 2000 (pp. 1685-1686). Chesapeake, VA: AACE.

[2] Harmon, J. & Marquez-Zenkov, K. (2003). Perpetual Pedagogy: A Critical Deficiency in Modeling Educational Technology to Pre- and In-Service Teachers. In C. Crawford, D. Willis, R. Carlsen, I. Gibson, K. McFerrin, J. Price & R. Weber (Eds.), Proceedings of Society for Information Technology and Teacher Education International Conference 2003 (pp. 3585-3588). Chesapeake, VA: AACE.

[3] Ahmad R., & Ives B., (1998) "Effectiveness of Virtual Learning Environments in Basic Skills Business Education: A Field Study in Progress.", with. Proceedings of the Nineteenth Annual International Conference on Information Systems (ICIS '98), Helsinki, Finland, December 1998.

[4] Hannafin, M. J., Hannafin, K. M., Land, S. M. & Oliver, K (1997).: Grounded Practice and the Design of Constructivist Learning Environments, Educational Technology Research and Development, 45(3), 1997, p.101-117.

[5] Hannafin, M., Land, S. & Oliver, K. (1999): Open Learning Environments: Foundations, Methods, and Models. In: Reigeluth, C. M. (Ed.). Instructional-design Theories and Models: A new paradigm of instructional theory, Volume II, 1999, p. 115-141. Lawrence Erlbaum.

[6] Affleck, G., & Smith, T. (1999). *Identifying a need for web-based course support*. Paper presented at ASCILITE, 1999, Brisbane,Australia. http://www.ascilite.org.au/conferences/brisbane99/papers/papers.htm

[7] Rogalski, J. & Samurcay, R., *Task Analysis and Cognitive Model as a Framework to Analyse Environments for Learning Programming* in Cognitive Models and Intelligent Environments for Learning Programming, Springer-Verlag publisher,1993

[8] Ben-Ari, M. (2001). Constructivism in computer science education. *Journal of Computers in Mathematics & Science Teaching, 20*(1), 24-73.

[9] Soloway, E. & Spohrer, J. (1989). Studying the Novice Programmer, Lawrence Erlbaum Associates, Hillsdale, New Jersey. 497 p.

[10] Stamouli I, Begum M., & Mancy R., ExploreCSEd: Exploring Skills and Difficulties in Programming Education

[11] Pea, Roy D.; Kurland, D. Midian, “On the Cognitive Effects of Learning Computer Programming”: A Critical Look. Technical Report No. 9.