



HAL
open science

Usability of virtual learning environment for learning java

Majlinda Fetaji, Suzana Loskoska, Bekim Fetaji, Mirlinda Ebibi

► **To cite this version:**

Majlinda Fetaji, Suzana Loskoska, Bekim Fetaji, Mirlinda Ebibi. Usability of virtual learning environment for learning java. Balkan Conference in Informatics (BCI 2007), 2007, Bulgaria. pp.523-534. hal-00190066

HAL Id: hal-00190066

<https://telearn.hal.science/hal-00190066>

Submitted on 23 Nov 2007

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

USABILITY OF VIRTUAL LEARNING ENVIRONMENT FOR LEARNING JAVA

MAJLINDA FETAJI¹, SUZANA LOSKOSKA², BEKIM FETAJI³,
MIRLINDA EBIBI⁴

¹CST Faculty, South East European University
1200 Ilindenska bb, Tetovo, Macedonia
e-mail: m.fetaji@seeu.edu.mk

²ETF Faculty, St. Cyril and St. Methodius University
Karpos 2, Skopje, Macedonia
e-mail: suze@etf.ukim.edu.mk

³CST Faculty, South East European University,
1200 Ilindenska bb, Tetovo, Macedonia
e-mail: b.fetaji@seeu.edu.mk

⁴CST Faculty, South East European University
1200 Ilindenska bb, Tetovo, Macedonia
e-mail: m.ebibi@seeu.edu.mk

It is widely stated that learning to program is very difficult in the traditional manner. To help novices learn object-oriented programming in Java more easily, we have developed the electronic Java virtual learning environment. In this paper, we discuss the usability of the learning environment and principles how to implement a usable learning environment. Considering the usability testing of the environment from the beginning stage of the development process, should improve the qualitative characteristics of the learning environment thus improving the abilities of users efficiently to use the environment, to be productive and fill satisfaction. We expect this way to make the learning process more efficient. According to the usability testing results we will give recommendations how to design similar usable learning environments.

Keywords: Usability, learning environments, java, graphical interface

1 INTRODUCTION

In the last two decades, computers and computer-based environments are widely used in higher education for teaching different subjects. In many cases high efficiency in education is accomplished. The development of highly interactive environments with advanced

graphical user interfaces (GUI) has become common. The new generation of these computer-based environments includes virtual learning environments. There are developed a number of virtual learning environments that offer learning content, services and learning framework to the users. But many of them contain a very complex user interface with many options hidden in multiple menu layers that are even noted as “user friendly”. So, the students may be lost in exploring the opportunities offered by the environment and may lose the goal of learning. This may be an obstacle in the learning process (Fetaji, 2007). Additionally, many of the environments are difficult for use and learn, difficult to remember and do not attract or keep learners. Several questions are raised 1) how to make simple, easy-to-use, more useful, and more productive environments; 2) how to provide users with experiences that fit to their specific background knowledge and objectives; and 3) how to improve the usability of the environment. These questions are usually not considered during the design process of the environment. In Section 2, we define the usability of learning environments and its importance. In Section 3 we discuss how to improve the usability of a learning environment. Section 4 contains the description of development of our learning environment considering its usability. In Section 5, to improve the usability of the system, we describe the necessity to extend usability testing adding qualitative issues to the testing using questionnaires. In conclusion, we present our recommendation how to develop a usable learning environment.

2 USABILITY OF A LEARNING ENVIRONMENT

A learning environment will be accepted or not from the users, according to its practical acceptability and usefulness that include its usability and utilization (Nielsen, 2003). If a website is difficult to use, people leave. If the homepage fails to clearly state what a company offers and what users can do on the site, people leave. If users get lost on a website they leave (Nielsen, 2003). In creating a successful online community, there are two critical components: sociability and usability (Nielsen, 2003). In creating a successful learning environment the critical components are: utilization and usability of the environment. Utilization of the environment can be defined as the extent to which the environment can offer what users need, or satisfies user’s needs (Nielsen, 1993). Usability is defined as the extent to which a system can be exercised to achieve specific goals and complete well-defined tasks effectively, efficiently and with satisfaction (Dix, 1998).

Usability is a qualitative attribute that assesses how easy user interfaces are to use (Nielsen, 2003). It is an attribute to measure the ease of use, the ease-of-learning to use, efficient and effective use and satisfaction of a user while using the system or in other words it measures the quality of user’s interaction to the learning environment.

Usability is defined by five quality components (Nielsen, 2003):

- Learnability: how easy users accomplish their basic tasks the first time they encounter the design.
- Efficiency: how quickly can users perform a task using the interface of the system?
- Memorability: how easy is to memorize how to use the interface of the system and how easily users can reuse the system after a break?
- Errors: How many errors do users make using the interface of the system and how serious are these errors?

- Satisfaction: How do users like using the system's interface?

The usability of the learning environment is an important attribute because it influences the user's satisfaction, the ability to learn and to remember the content of the learning environment; in minimizing number of errors during interaction with the leaning environment which leads to more effective and efficient learning. There are some potential benefits of a usable learning environment: 1) More efficient user tasks lead to better and more effective learning activities, 2) More effective user activities lead to more effective and successful learning, 3) Increased satisfaction leads to better learning, enjoying interaction with the environment, 4) Increased satisfaction and effectiveness lead to reuse of the learning environment, 5) Reduced errors lead to more effective use of the environment and greater satisfaction, 6) Users are satisfied, not frustrated, and achieve their goals more effectively, 7) If the system is usable, it is easy and efficient for use, easy to be learned, and users make few errors using the system while they fill subjective satisfaction (Nielsen, 1993).

3 THE CONCEPTUAL DESIGN OF JAVA VIRTUAL LEARNING ENVIRONMENT

The system is designed on the hypothesis that the system's interface is crucial towards improving learning a programming language. The complexity of the GUI (graphical user interface) of the today's most popular JAVA editors or IDE' s is usually discouraging for a lot of novices and they need a considerable amount of time to understand it, and use it Shneiderman, B., (1986), so the effectiveness of learning to program might be lost or redirected. Also, they need to be installed and usually are not distributed without any change. On the other hand, this project will have a simple GUI and be easy to understand it and use it, and will be distributed for free download and the more important is the system doesn't need to be installed, meaning we minimize the needed requirements to a simple run of an application. And since it will offer just the essential functions needed to write Java code providing an editing environment, to compile it using the Sun Javac compiler, and to run the Java code, and getting back the compiling errors if any exist or getting the result of the program execution without leaving the system's framework, it will allow the users to concentrate on the language structure and the principles of coding. This will help students to focus on the language structure and programming concepts and learn it by typing and creating all needed elements by their own.

Observing that all novices and advanced programmers usually learn to program by using and observing different existing programs or program modules in books, manuals or online examples, our second objective was to develop a tool implementing the approach to provide program examples within the development environment integrated in the help component and also to provide three other kinds of help: application usage help with introduction help, application help, and Multimedia help which covers the usage of the application and all the documentation embedded in the application. Then the links to Java online tutorials, that will cover the Java language in general, and finally the online support which offers the frequently asked questions, feedback forms and email contacts for communication with the user and higher interactivity. All this is thought as an e-learning environment that allows the users having everything they need in one place and have faster progress in learning (Shneiderman, B., 1986).

To successfully create a program, users must understand several issues: how to express instructions to the computer (e.g., syntax), how to organize these instructions (e.g., programming style), and how the computer executes these statements (Nielsen, 2003).

To facilitate this issue, we developed useful components to the system that will generate a design Java template and Java Applet codes. Our goal was to add some programming constructs to the system. "An environment should conform to the programmer, rather than force a programmer to make his or her concepts of how a system should work conform to an environment" (Reiss 1986); therefore, we thought that our system would be a better tool for learning Java programming than the existing popular complex GUI editors by providing a simple GUI that does not require any installation or any previous computing knowledge in order to use it. Also, by providing examples, tutorials, design templates, that will facilitate the processes of software development. The technical issues in terms of minimized hardware and software requirements also are an advantage of our system and facilitate the usage of the system.

Another advantage is that students teach how to discover information and generate knowledge as a result of their interactions (Nielsen, J. 1994) with the learning environment, like the help examples, tutorials and then testing and changing the code of the provided examples to test different concepts and logic of the module by quick interaction to compile and run buttons of the interface. Also, they can interact to the avatar to listen the help contents related to learning tutorials or using the system.

4 THE CONCEPTUAL AND FUNCTIONAL ARCHITECTURE DESIGN OF THE LEARNING ENVIRONMENT

The conceptual and functional architecture design of the Java learning environment is shown in Figure 2.

5 ISSUES OF DESIGNING A USABLE LEARNING ENVIRONMENT

The basic thinking of developers is I am developing for humans; I am a human; I can make something that is good for humans (Evans K.M. & Finck N, 1993). But, that is the biggest mistake, because the developer knows how to develop and program the functionality of the system and what the system offers. Other questions like how the system can do and whether the system is usable, are usually omitted. There is a need for incorporation of users in the design process because users have an infinite potential for making unexpected misinterpretations of interface elements and to perform their job in a different way than you imagine (Nielsen, 1994). To develop good software, qualitative methods for its usability testing are required. The question raised in the introduction of the section can be deceived using user-centered design involving students in the development process (Evans K.M. & Finck N, 1993). The usability of an environment is fundamental for successful learning. If the learning environment has: i) difficult-to-use design, ii) a design that is difficult to be learned to use; iii) a design difficult to remember how to reuse; iv) content structure and organization that is a mass; v) workflow in the process of using that is difficult to be comprehended; users will not be effective, efficient and productive and will feel scared, frustrated and dissatisfied

while understandability and learnability of a subject may suffer. They will not want to use it again. So, the users need a usable learning environment. We need to use a corporate policy to promote usability, enforce standards, and fund user-centered design that is the only way to quality user interfaces (Evans K.M. & Finck N, 1993). The interface must be efficient and easy-to-use, so the user will concentrate on the learning goals, learning content and activities not in learning how to use the environment. Understanding how learners interact with the learning environment using its interface will contribute in designing efficient and easy-to-use interfaces which then will contribute to improve the decision-making process and enhancing the quality and effectiveness of learning. Using design guidelines are crucial in the design process of a usable learning environment. But, the best way to ensure usability is testing users. The most effective method for highly usable system is simple user testing: get four to five real customers to sit down by a computer (one at a time) and use your design while they think out loud (Evans K.M. & Finck N, 1993). With the usability testing method we want to ensure that we will get best critics for the usability of the system that intended users can carry out the tasks efficiently, effectively and satisfactorily.

6 DESIGN-PROTOTYPE-EVALUATE CYCLE

To design our system we have followed the “design-prototype-evaluate” cycle that means we have designed the interface based on results obtained by users’ testing, by observing users’ interaction and on the qualitative empirical research conducted in each iteration (Fetaji, 2007). In the interface design process, we have employed the spiral method life-cycle of the design as suggested in (Evans K.M. & Finck N, 1993). According to Jacob Nielsen’s suggestion, a strategy for usability in design includes the following steps (Nielsen, J. 2000):

1. run a small user test on the old design
2. test mocked-up (paper or static graphics) prototype designs on users
3. develop the best prototype and test with more users
4. iterate the design as often as possible (test, fix, test, fix etc)
5. do a final test
6. work on the next design

A. Design

We have used the User-Centered Design, because the results of the design decisions are unknown. The User-Centered Design takes the unknown out of the design; you can see exactly how users will use it. The Design is based upon:

- user’s needs,
- user’s cognitive abilities (vision, perception, action, memory, reasoning),
- user’s educational background (previous education, knowledge, computer skills) (Jones at al., 2003),
- tasks to be achieved,
- context of tasks (Dix, 1998) (characteristics of tasks, how are the results of task performing controlled, dependencies between the tasks (Workflow), share of power and influence, existence of a technical support),

- learning outcomes.

As stated above, the first version of the user interface was drawn on a paper and is set to comply with the Nielsen's (Nielsen, 2000) nine basic usability principles:

1. Use simple and natural language.
2. Speak the user's language.
3. Minimize user memory load.
4. Be consistent.
5. Provide feedbacks.
6. Provide clearly marked exits.
7. Provide shortcuts.
8. Provide good error messages.
9. Prevent errors.

The final version of the interface of the virtual learning environment Java is shown on the Figure 1:

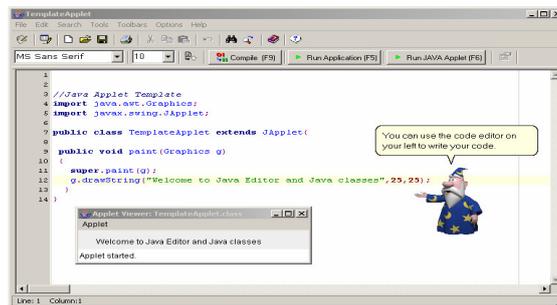


FIGURE 1. GUI of the virtual learning environment

To design the HCI properly, we have also partly followed the Shneiderman's strategies for effective human computer interactions (Shneiderman,1986). We also used the general principles and guidelines for HCI regarding the software design from (Pressman, 2005), and general principles and guidelines for document design and guidelines for online documentation (Dumas, J. S., & Redish J. C. 1999). All this guidelines were closely advised and reviewed when designing the system.

B. Involving users in each stage of the design

In the design we have involved users from the beginning and in each stage of the prototype design until the implementation of final design..Prototyping is a step iterated in the process of the user-centered design of an interface where we got information from users about the design, possible errors; difficulties in the interactions and is any improvement in the design possible before the final implementation of the design.

1. First stage of the design

Employing spiral method of a software life-cycle, we designed the initial design without testing. The initial look was a Windows application, more adjacent to application style that students were used to, according to their background experience. Objects on the interface that represented tasks, options, activities and utilities and the learning content help

menu were represented by words. The developing environment was represented as an empty white box.

The following prototypes were designed on a technological platform known as high-fidelity prototypes, because those are useful in testing out technical issues (Preece, 2002).

2. Second stage of the design

User' testing and observation results indicated that:

- They like this simple Windows style look of the environment;
- They prefer more toolbar buttons for frequently used functions and also a menu representation of all functionalities of the environment;
- They prefer an image related to the functionality to be added to each button, because they remember easily the visual representation, and a hint that would remind about the functionality;
- They would like some templates to Java codes and applets to increase the speed of writing the code;

Observing users' interactions, we encountered difficulties in reading the fonts and the size of the opened document. Some users were confused attempting to change the color and the size of text. Also, some of the users were confused when attempted to use keyboard shortcuts for the most frequent operations like opening help file or program code, and "compile-run" sequence. They would prefer using shortcuts for quicker performance.

The experts' remarks on the first prototype were on the long terms used to specify different objects and the number of the elements on the interface was large. The second prototype was designed based on these results.

3. Third stage of the design

User' testing and observation results indicated that:

- They like the idea to add a Microsoft avatar to support learners with audio-visual effects.
- They would like to have more help content and examples.
- They would like more error messages and provided help to fix errors.

The experts' remarks on the second prototype were about error messages which should be shorter and more meaningful, and prevention from errors that were not supplied. The third prototype was designed according to these results. We have better results from measuring usability attributes of the third prototype published in (Fetaji, 2007).

C. Testing and Evaluation Strategy

In the research literature user testing is considered as the most useful. To measure the usability aspects of a system's user interface (UI) and to identify specific problems, besides user testing, the usability evaluation is used. It is an important part of the overall user interface design process, which consists of methodologies for measuring usability aspects and iterative cycles of designing, prototyping, and evaluating (Dix et al, 2003), (Nielsen, J., 2002).

As suggested in (Evans K.M. & Finck N, 1993), "think out loud" method is not appropriate for this kind of environment and does not always work well, because the extracted information is not relevant, since we have to count on students feeling afraid and as they are taking an exam.

Besides the quantitative study of design by observing real users how they interact with the system and empirical research study of measuring the usability attributes defined by (Fetaji, 2007): time to learn, speed of performance, rate of errors (in using the interface), subjective satisfaction; the evaluation of the system requires a qualitative study, we suggest using a questionnaire. The quantitative testing results may not be very relevant and can be misleading because of the individual differences among individuals (Nielsen, 2006). Users with different background experience can give us quantitative feedback of variable relevance which influences the quality of overall design. Quantitative studies are based on statistical analyses that, as Jakob Nielsen states are often false, biased, misleading, or overly narrow. Better to emphasize insights and qualitative research (Nielsen, 2004). By observing real users we can faster detect the issues, and get much information, but the observer influences on the conclusions. Using questionnaires, gathering opinions and understanding of the users brings more qualitative data. And both methods cost time and recourses. The number of users sufficient for the testing is five as is stated in (Nielsen, 2003), but in quantitative testing it is recommended to test 20 users (Nielsen, 2003). We need to incorporate different users in the testing because there are tremendous individual differences among users (Nielsen, 2006). (Nielsen, 1993) suggested that heuristic evaluation and user testing should be alternated, because these two usability practices have been shown to find fairly distinct sets of usability problems. Also, the heuristic evaluation is the most general of the usability inspection methods and is also the easiest to learn and apply (Christie M., 2007). Thus, besides the users testing, the heuristic evaluation is conducted according to Nielsen's ten usability heuristics (Nielsen, 1994) to find many usability problems that are overlooked by user testing, where a group of experts are invited to examine the user interface for demerits and whether each dialogue element follows the established usability principles according to (Christie M., 2007). Experience from many different projects has shown that different people find different usability problems. Therefore, it is possible to improve the effectiveness of the method significantly by involving multiple evaluators (Nielsen J., 2006). In this way. we will ensure and generate important quality improvements in the system's usability.

Users and tasks:

We have conducted user testing according to (Nielsen, 2003). Therefore we have chosen representative users, students that are intended to use the system. The representative users were students in Computer Sciences. We have defined two types of users, novices and experts. All Users have been set to have full access to all options.

We asked the users to perform representative tasks that are important in learning programming and also practicing programming as a component of learning, with the design. To discover possible errors and difficulties in achieving learning goals the representative tasks were those to be accomplished in the final solution as simply and effectively as possible:

- Open a Java source code file
- Write and view the code
- Save the code to a file
- Compile the code to a Java byte code (compiled class)
- Execute (run) Java application
- Execute (run) Java applet
- Validate syntax programming code errors and present it to the user

- Using the help component for learning goals

We have measured the usability attributes while interpreting representative tasks with the design (Fetaji. M At al., 2007). We have observed how the users interact with the interface, where they succeed, and where they have difficulties with the user interface. We did not provide help for users during their work with the system. We just left them on their own to start using it.

Users' requirements:

To meet the users' requirements for learning to program in Java, the learning environment should provide the following:

- Operations for functions needed to test and practice programming concepts (as an important component of learning programming)
- Operations for manipulating with code files
- Shortcuts for most frequently used operations and templates
- Learning content for learning programming concepts
- Help content for the usage of the environment

7 EXTENDED USABILITY TESTING USING QUALITATIVE APPROACH

Quantitative methods of usability testing by measuring usability attributes such as learning time, efficiency of use, user errors, and user satisfaction are very expensive and hard to conduct. Besides the bigger number of users required, it is also hard to gather users in one place and measure the attributes, and it is necessary a systematic recruiting to actually represent the target audience (Nielsen, 2006). As suggested in (Nielsen, 1993), issues related to attributes of usability, moreover subjective satisfaction and possible anxieties can be best detected by asking users. To get qualitative information about issues related to usability attributes we decided to do an extended qualitative study of the usability of the environment. To improve the usability of the learning environment by getting qualitative feedback from the users, we have followed the approach of testing usability using a questionnaire. We have grounded the design of a questionnaire on Nielsen's basic principles of usability (Nielsen, J., 2000). Some of the questions in the questionnaire are presented in Appendix A.

A. Usability testing of the interface using questionnaires

Users and tasks are defined in Section 4.2. For this qualitative study we choose 8 users, 5 of them considered themselves experts and 3 novices. We did not provide help for users during their work with the system. First we left them on their own to start using the environment. After using it, we gave them to fill the questionnaires. The results that we got are as following:

The questions that had been answered narrowly indicated that:

They have used this kind of application,

They do not need to spent much time to learn to use the application

They learn quickly

They understand how to use the system easily

They have liked the implementation of "edit-compile-run" cycle

They have indicated they do not use shortcuts frequently

They think students will benefit from using it in time, speed to perform a task

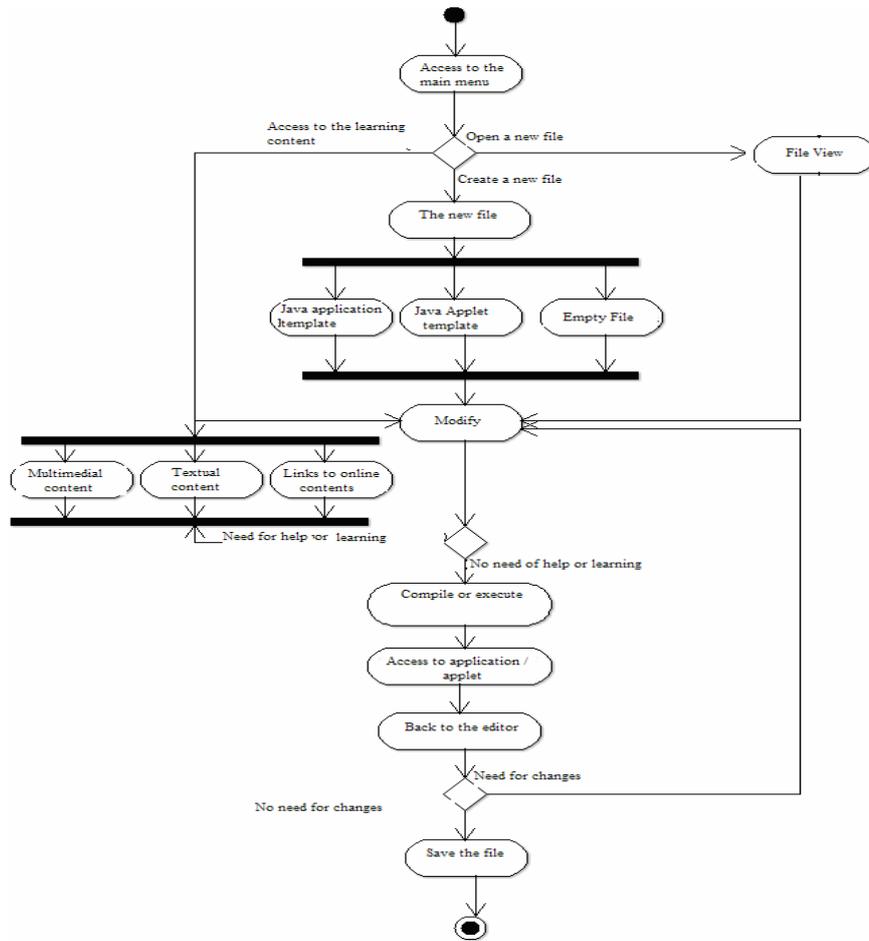


FIGURE 2. The conceptual and functional architecture design of the JAVA learning environment

They are willing to use it in a future

Experts indicated that this environment is simple and they would like to use more advanced environment

8 CONCLUSION

From the results of the usability testing of the design using quantitative methods for users' testing we improved the design of the virtual environment for learning Java (Fetaji at al, 2007). To complement the need for qualitative data, we made extended users' testing using questionnaires. According to the results of the testing, we had more qualitative data using this

approach. We can conclude that the strategy of designing a usable learning environment requires a qualitative approach to best gather qualitative information about the usability of the environment, besides its benefits. Thus, we will ensure to get critical information to improve the design of the interface before its final implementation.

In designing our learning environment, of great importance was the user-centered design approach where we included users in the design from the beginning. That was very helpful in designing the interface. In this way, we ensured that it will satisfy the users' needs, correspond to users' abilities and background, enable efficiently to perform tasks and achieve learning outcomes. Also, adding the qualitative approach to users testing would improve and help the design process. To improve usability of a similar learning environment, we can summarize recommendations as follows:

- Involve users in the design process from the beginning stage and use user-centered design,
- Use combination of the quantitative method of users' testing by measuring usability attributes and the qualitative method by questionnaires; and professionals' evaluation in each stage of the design for getting quantitative and qualitative data. So, you will best gather qualitative information about the usability of the environment. We recommend similar human-computer interaction design to similar solutions.

REFERENCES

- 1 Nielsen J., Usability 101 (2003): Introduction to Usability, Alertbox, August, 2003
<http://www.useit.com/alertbox/20030825.html>
- 2 Nielsen J., & Mack L.R.(2004),Usability Inspection Methods, John Wiley & Sons, Inc., 1994, New York
- 3 Nielsen J., (2000) Designing Web Usability: The Practice of Simplicity. New Riders Publishing, Indianapolis, ISBN 1-56205-810-X, 2000
- 4 Nielsen, J. (1994) Usability Engineering, San Francisco. ISBN 0-12-518406-9
- 5 Preece, J. (2005),Online Communities: Designing Usability, Supporting Sociability. Chichester,, John Wiley & Sons.
- 6 Dix, A., Finlay, J., Abowd, G. & Beale, R. (1998) Human-Computer Interaction. 2nd. edition. UK: Prentice Hall
- 7 Nielsen, J. (1993). Usability Engineering. Boston, MA: Academic Press.
- 8 Evans K.M. & Finck N (2003)., Interview with Jakob Nielsen, The web professional's online magazine of choice
- 9 Nielsen, J., (2002) "How to Conduct a Heuristic Evaluation".
http://www.useit.com/papers/heuristic/heuristic_evaluation.html
- 10 Jakob Nielsen. (2001) " Summary of Usability Inspection Methods"
http://www.useit.com/papers/heuristic/inspection_summary.html
- 11 Nielsen J., (2006) Variability in User Performance, Alertbox, May, 2006
http://www.useit.com/alertbox/performance_variability.html

- 12 Christie M., (2007) “Cognitics”, Master Course - SEE University
- 13 Fetaji M., Fetaji B., Ebibi M., (2007)“Modeling e-learning virtual environment for learning java programming” – Accepted and to be published in the proceedings of the International E-Learning 2007 conference, Lisbon, Portugal, 06-08 July 2007.
- 14 Jones, P., Packham, G., Miller, C., Davies, I., Jones, A.(2003) “e-Retention: An Initial Evaluation of Student Withdrawals within a Virtual Learning Environment”, Proceedings of the 2nd European Confrence on eLearning, Ed. Roy Williams, Academic Conferences International, Reading (2003) pp.239-248.
- 15 Shneiderman, B., (1986). Designing the User Interface: Strategies for effective human-computer interaction. Addison Wesley, Reading, Massachusetts.
- 16 Nielsen J. Quantitative Studies: How Many Users to Test? Alertbox, June, 2006
- 17 Nielsen, J. (1994) “Ten Usability Heuristics”,
http://www.useit.com/papers/heuristic/heuristic_list.html
- 18 Dumas, J. S., & Redish J. C. (1999) “A practical guide to Usability Testing” revised edition, Pearson Education Limited, pp.55-62
- 19 Pressman, R. (2005) “Software Engineering: A practitioners approach 6 Ed” McGraw-HILL, inc, pp.81-117
- 20 Preece, J., Rogers, Y. and Sharp, H. (2002) Interaction Design: Beyond Human-Computer Interaction, John Wiley & Sons, New York
- 21 Nielsen J., (2004b) Risks of Quantitative Studies, Alertbox, March, 2004
<http://www.useit.com/alertbox/20040301.html>

Appendix A

	Question	Answer:
1	What kind of user do You express yourself?	
2	Have you ever used this kind of application?	
3	Can you explain what is this application for?	
4	What did you mostly like when using this application?	
5	What did you not like when using this application?	
6	What can be improved in this application?	
7	How much time did you need to learn to use this application?	
8	Are there any shortcuts supplied in the application?	
9	How easy to understand are the shortcuts?	
10	How often do you use shortcuts?	