Usability Criteria for E-learning Virtual Labs

Orlin Tomov

Abstract: *The paper describes some aspects of the testing methodology, used to approve the virtual labs, developed for the "Computer Organization" and "Embedded Systems" courses in the University of Ruse.*

Key words: *Computer Systems and Technologies, Testing Software, Computer Organisation, Simulator, Embedded Systems.*

INTRODUCTION

Along with the development of virtual labs in the University of Ruse, arose the question about testing and approving usability of the software, used for it. A methodology for testing on system level is already developed in the department [5], while this article focuses on the testing of the user level and user interface. Easiness of use affects the users' performance and their satisfaction, while acceptability affects whether the product is capable for use [3].

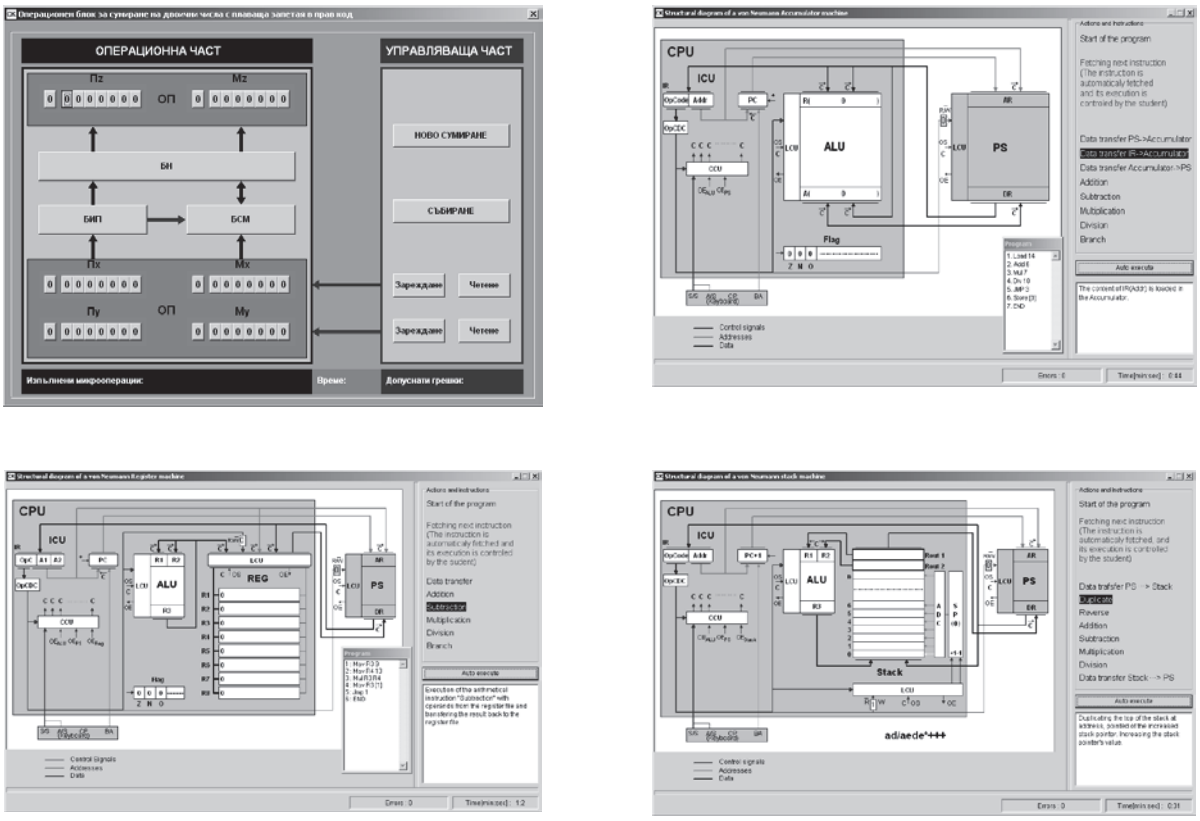

Fig.1. Simulators, used in the virtual lab of the "Computer Organization" course

Since different applications with different organisation have been built, different testing algorithms and strategies should be used. For instance – the virtual lab of "Computer Organisation" is based on simulators of different microprocessor modules and blocks, while the lab of the "Embedded Systems" course is based on a system for remote access to development boards. On another hand the two labs are characterised with different targets. The first (Fig.1) is focused on the demonstration and learning on the principle of try and fail, but the second lab is based on remote access and remote experiments with the equipment.

If the implementation is not given a sufficient attention, the e-learning process will be unsuccessful. In fact, while many companies are experienced with e-learning services, others have aborted their efforts [1].

Some of the most important keystones are marked and described in this article, during the evaluation of the virtual labs on the two courses was performed

DEFINITION OF THE COMMON TESTING PARAMETERS

The testing procedures` first step starts immediately with the development. Along with the source code writing and compiling, many errors on system level have been found and corrected. This is the so called "formative" testing [6]. At a later stage, we continue with the "summative" testing with its all aspects.

The following qualities are required to define a system as usable [4]:

- Effectiveness – the performance the system should poses;
- Flexibility – adaptation to variable tasks;
- Users` satisfaction – the system has to satisfy the user's requirements

We can define the requirements for our labs on the base on the following qualities:

- Technical requirements:
  - Compatibility with different software platforms – this means that the software, developed for the virtual lab, should be able to run on different platforms. For example – Windows, Linux, MacOS.
  - Easy installation. If the installation of the software is complicated, and requires specific system settings, compilation, or installing additional packages, this may result negative of the virtual lab usage and popularity;
  - Extendibility and possibility for future upgrades. Usually the functional requirements of the software product are not constant all the time. Soon after the finish of the development, new functionalities are required. In order to allow easy update and upgrade, is needed the architecture to be extendable or even modular. This feature also decrease the cost of the product in a long-term period, since implementing new functions does not require redesigning and re-engineering of the whole project;
  - Simple user interface – the user interface should be simple enough to allow intuitive work with the lab, and complicated enough, to allow the representation of the subject. If the product requires time-to-learn for its interface, this could distract the learners and to place their attention on the use of the lab, instead on the objects, represented by it. Thus, the development team should find a balance, which could be achieved with several iterations of redesigning the interface, and testing with real users.
  - Memory consumption, with its both aspects – disk space and RAM. Disk space is important, since the huge applications are not preferred for download and installation by most of the users. Usually this could be achieved by realisation of simple algorithms and always leads to stable functioning. On another hand, high RAM memory usage requires an expensive hardware that could be also a barrier for the end-users.

- Environment related requirements
  - Subject area for the software product. This should be selected very precisely. If the area is too large, the students will not be able to

understand the simulation, and if it is too small – the results will be too obvious and the effect of this product will be not as desired;
- Abstract level of presentation. Just like in the above point – there should be a balance between high and low abstract level – if it is too high, the student will not understand the actual principles of work of the represented object. If it is too low, the complexity could be a huge barrier;
- Target group of users.
- End-user knowledge and skills on the subject area;
- End-user ability to work with software products. It is an importance to define the target group of users for the last three points. The developer must be familiar with the organisation of the interface, abstraction level of representation, documentation and teaching materials, based on the virtual lab. This corresponds to the social usability group criteria [2].

In general, help documents and other materials, related with the lab, have to be avoided. This means that the interface must be as intuitive as possible. The represented processes and functions have to be as close as possible to the teaching material of the course. The students should use only the course materials for their work with the virtual laboratory. Otherwise, if they have to familiarize themselves with the tools, or to learn how to work with it, this will consume time and will focus the teaching process on the laboratory organisation. This is an unwanted effect and should be avoided at any price.

SPECIFIC NOTES ABOUT THE COMPUTER ORGANIZATION COURSE LAB

Some of the software tools, used for this laboratory are presented in Fig.1. These are used for the representation of the physical organisation of the different processor modules and their work. At the screenshots are presented the main screens of the following tools:

- Adder for floating point numbers;
- Accumulator von Neumann machine;
- Stack von Neumann machine;
- Stack von Neumann machine, version 2;

The tools contain a schematic view(s) of the described block or module with clickable fields for the clock signals. The student plays the role of the management unit and has to determine the correct sequence of "clock signals", to complete the task, given by the lecturer, or to experiment with his own one. In the first versions of the tools, there were only short descriptions of the required microoperations, supporting the student.

Our first tests with students did not represent the expected results. That was on the ground of the amount of information, which was fed to the students [7]. Their inability to orient themselves in the new environment as fast as we expected required some changes to be made. The simplest method we found to escape this situation was adding a "demo" button, called "Auto execute", which executes automatically the next needed microoperation.

The tests with the second versions showed up much better results. While the familiarizing process of the students with the tools required 15-30 minutes, the second version decreased this time to 5-15 minutes.

There is also a hidden help subsystem that counts the students' mistakes and wrong microoperations and provides more information for the next step according to the number of mistakes.

SPECIFIC NOTES ABOUT THE REMOTE ACCESS SYSTEM

In general, all the tips, mentioned above are valid for the remote access system (Fig.2) on the "Embedded Systems" course. Because of the specifics of the client-server systems, some additional requirements should be defined.

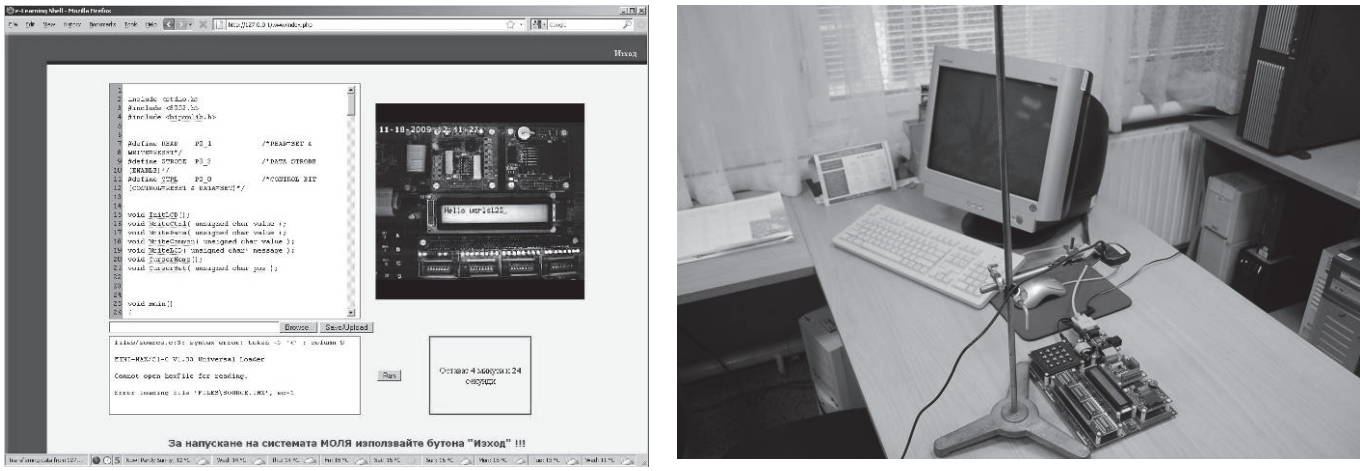

Fig.2. Remote access system, used for the "Embedded Systems" course

These are:

- Bandwidth – when it is too high, it could be not possible for some students to access the system. The bandwidth is related with the speed. There are no industry standards for that. Everything depends on the given situation and process ;
- Required additional software - like frameworks, browser plug-ins, etc. The best case if nothing additional, than the standard configuration is required. In some cases this is not possible, so the development team should use most popular ones (such as Java, .Net, Flash), because in most of the user software configurations they are almost always present, and the installation of the client software has no additional requirements or dependencies.
- Feedback – in the case of a remote access system, the feedback is crucial. Any delay or information loss may cause a failure of the experiment.

CONCLUSIONS AND FUTURE WORK

The work on this problem in the department of computing should continue with a methodology for measurement and assessment of the parameters, described in this proceeding.

A detailed questionnaire for feedback from the students should be also developed in the future.

REFERENCES

[1] Howard, B. (1998), Increasing employee knowledge and understanding of operational systems:Integrating multiple technologies at NYNEX. In D. A. Schreiber and Z. L. Berge (eds), DistanceTraining: How innovative organizations are using technology to maximize learning and meet business objectives (pp. 92–114). San Francisco.

[2] Bevan, N. Usability is Quality of Use, Proceedings of the 6th International Conference on Human Computer Interaction, Yokohama, July 1995. Anzai & Ogawa (eds), Elsevier

[3] Bevan, N. Measuring usability as quality of use. Softw. Quality J. 4 (1995), 115–130.

[4] E Folmer, J Bosch. Architecting or usability: a survey. Journal of systems and software, 2004 - Elsevier

[5] Vasileva, A. Usability Evaluation of a Virtual Laboratory on Computer Organization. Proceedings of Rousse University Annual Conference, 31.10 -1.11 2003, pp. 175-180.

[6] Barnum, C. Usability testing of e-Learning: Easy, Effective, Affordable. eLearn Magazine , March 2008, http://www.elearnmag.org/subpage.cfm?section=best_practices&article=47-1

[7] CM Chen, HM Lee, YH Chen. Personalized e-learning system using Item Response Theory. Computers & Education, 2005 - Elsevier

ABOUT THE AUTHOR

MSc. Eng. Orlin Tomov, Department of Computer Systems, University of Ruse, Phone: +359 82 888 276, E-mail: OTomov@ecs.uni-ruse.bg

This paper is financed by project: Creative Development Support of Doctoral Students, Post-Doctoral and Young Researches in the Field of Computer Science, BG 051PO001-3.3.04/13, EUROPEAN SOCIAL FUND 2007–2013г. OPERATIONAL PROGRAMME "HUMAN RESOURCES DEVELOPMENT"