Georgiadou, E., Economides, A.A., Michailidou, A. & Mosha, A.: **Evaluation of online educational software designed for the purpose of teaching programming.** *Proceedings WebNet2001 World Conference on the WWW and Internet*, pp. 399-404, AACE 2001.

Evaluation of Online Educational Software Designed for the Purpose of Teaching Programming

Dr. Elissavet Georgiadou
Teacher in Secondary Education, Researcher
University of Macedonia, Greece
elisag@otenet.gr

Dr. Anastasios A. Economides
Ass. Professor of Computer Networks
University of Macedonia, Greece
economid@uom.gr

Anna Michailidou
Teacher in Secondary Education, Postgraduate Student
University of Macedonia, Greece
michan@uom.gr

Anna Mosha
Computer Analyst, Postgraduate Student
University of Macedonia, Greece
mosha@uom.gr

Abstract: In this paper we evaluate educational software, available in the market, designed for the purpose of teaching programming that addresses mainly beginners and intermediate students. The evaluation is conducted on the basis of an evaluation model built in University of Macedonia, Greece in the context of project "Odysseia" that is mainly designed to introduce computer and network technologies into the everyday school practice, especially in secondary education. The paper discusses briefly the project and introduces the educational software that was evaluated. Next, it presents the evaluation method and the processing of the data and finally reports some conclusions derived from the evaluation.

Introduction

The number of products from the educational software industry has significantly increased the last decade. The reason for that increase is closely related to the basic policy assumption that educational system should serve the overall target of 'information society for all' (see CEC 2000). Schools should prepare students to use actively new information and communication technologies (ICTs) taking advantage of the life-learning process that these technologies support. As a result in most countries all curriculums are under continuous development adopting ICTs in teaching and learning. In Greece the national curriculum is currently under reform, especially in secondary education, attempting to follow educational and technological advances. A number of projects come with this reform. One of the biggest in that respect is 'Odysseia' Program, which is part of the Ministry of National Education and Religious Affairs' Operational Program 'Educational and Initial Vocational Training' and is implemented by the Directorate of Studies for Secondary Education, the Directorate of the Community Support Framework, the Pedagogical Institute and the Computer Technology Institute, which are underwriting the design and technical support and monitoring the implementation of the 19 Projects that make up the Program. It is a dynamic, action-oriented program designed to cultivate and develop the faculty of critical thinking and to change the practices of teaching, learning and communication in Greek schools through the use of computer and network technologies in secondary education. These Projects are designed to introduce computer and network technologies into the everyday school practice of more than 370 secondary schools in Greece in every subject on the standard curriculum, in order to create a substantial number of school communities which have incorporated these technologies as an integral part of their daily teaching and learning practice. Project E42 is a particularly important facet of the overall endeavor to inspire in teachers a positive attitude towards the use of the new technologies, while at the same time educating teachers to become capable users of these technologies. The proposed training programs are addressed to teachers of all school subjects, not only computer science. The special one-year post-graduate training programs carried out in the modern Instructor Training Units at selected Greek universities are training a nucleus of 100 selected teacher/educators who will thus be able to provide in-school training for their approximately 6000 colleagues in the Odysseia schools. In that respect the University of Macedonia in Thessaloniki runs the action "EPENDISI" which not only aims to train secondary schoolteachers but also to build a database that contains information and resources on several evaluated educational software available in the market on almost all secondary school subjects (www.ithaca.uom). The evaluation of the software is conducted on the basis of an evaluation model built in University of Macedonia, that examines both social and practical acceptability of educational software (see Georgiadou & Economides 2000). This paper will present the evaluation results on eight online educational software designed for the purpose of teaching programming that address mainly beginners and intermediate students. In the following section, we briefly present each one of the programming software.

On-line Programming Software Presentation

An Introduction to MSW Logo: www.southwest.com.au/~jfuller/logotut/menu.htm
It is a thorough tutorial concerning the programming language MSW Logo. It covers basic programming aspects such as simple commands, the repeat command, functions, variables and some additional features. It includes instructions and suggestions for teachers and additional links concerning Logo.

How C Programming Works: www.howstuffworks.com/c.htm
This is a tool for Web authoring and it is compatible with any browser. Works with self-contained page sets instead of separate pages, offering an easy to use, fast environment to edit sites. It includes tables, maps, frame sets, slides, forms, windows opening, style-sheets and more - all in WYSIWYG.

Introduction to C Programming I: www.cit.ac.nz/smac/cprogram/cstart.htm, http://www.studyweb.com/
This is an in depth tutorial covering the C programming language. It assumes little or no knowledge of the C language, but does assume some knowledge of programming in general. It includes interactive multiple-choice exercises that can help the students to study and understand the issues presented.

Introduction to C Programming II: devcentral.iftech.com/learning/tutorials/c-cpp/c/
This series contains a set of tutorials that help one to learn about the C programming language, which start with the assumption that the user know a procedural language like Pascal or Fortran already, and simply want to map that knowledge to C. These tutorials introduce the user to C by showing him/her how Pascal maps to it. It also introduces several concepts not found in Pascal. Most of these new concepts deal with pointers. Readers coming from a Fortran, Cobol, BASIC, etc. background should find that the Pascal code is easy to read.

Logo: library.thinkquest.org/18446/eintro.shtml
It is a tutorial that concerns the Logo programming language. It includes 10 lessons with the basic aspects of programming, such as simple commands, the repeat command, functions, etc. It also includes examples and exercises that can be modified by the students.

LOGO Programming: members.aol.com/mckoss/logo.htm, http://www.studyweb.com/
This is an online small tutorial for learning LOGO a child's programming language. This tutorial is designed for an adult to assist the child with the learning process.

Pascal Programming v2.1: www.mgu.bg/manuals/pascal/default.htm, www.cit.ac.nz/smac/pascal/default.htm
This is an online courseware for beginning Pascal programming. It starts with a discussion of variables and ends with an online interactive test. Every subject contains exercises that can be done interactively.

*rLogo***:** www.embry.com/rLogo
rLogo is an easy to learn programming language designed for the World Wide Web. It is based on the Logo programming language.

Evaluation

Evaluation Method

For the purpose of the evaluation of educational software a framework has been designed based on the integration of a number of important issues emerged from research on instructional design and system evaluation the past fifteen years, and which should be considered from designers and evaluators of hypermedia courseware. The framework is concerned with both social and practical acceptability of educational software, based on Nielsen's idea that "the overall acceptability of a computer system is a combination of its social and practical acceptability" (Nielsen 1990). The term social acceptability is related with the social basis of an educational system. In cases when the basis is teacher-centered, then the software that provides high levels of learner control is possibly socially unacceptable and vice versa. Given that a piece of educational software is socially acceptable, its practical acceptability is examined through the evaluation of the following sectors: Content, Presentation and Organization of the Content, Technical Support and Update Process, and finally, the Evaluation of Learning. All sectors are equally important, as educational software has to be simultaneously pedagogically and technically sound. Moreover, each sector includes a number of criteria, which should be meet in a satisfactory level, in order to characterize a piece of educational software of high quality. Furthermore, cost-effectiveness is always being examined when similar products seem to have the same educational value. The following figure (Fig. 1) presents all factors included in the framework only in a diagram due to the lack of space. However, all the details about it can be found in "Evaluation Factors of Educational Software" (Georgiadou & Economides 2000).

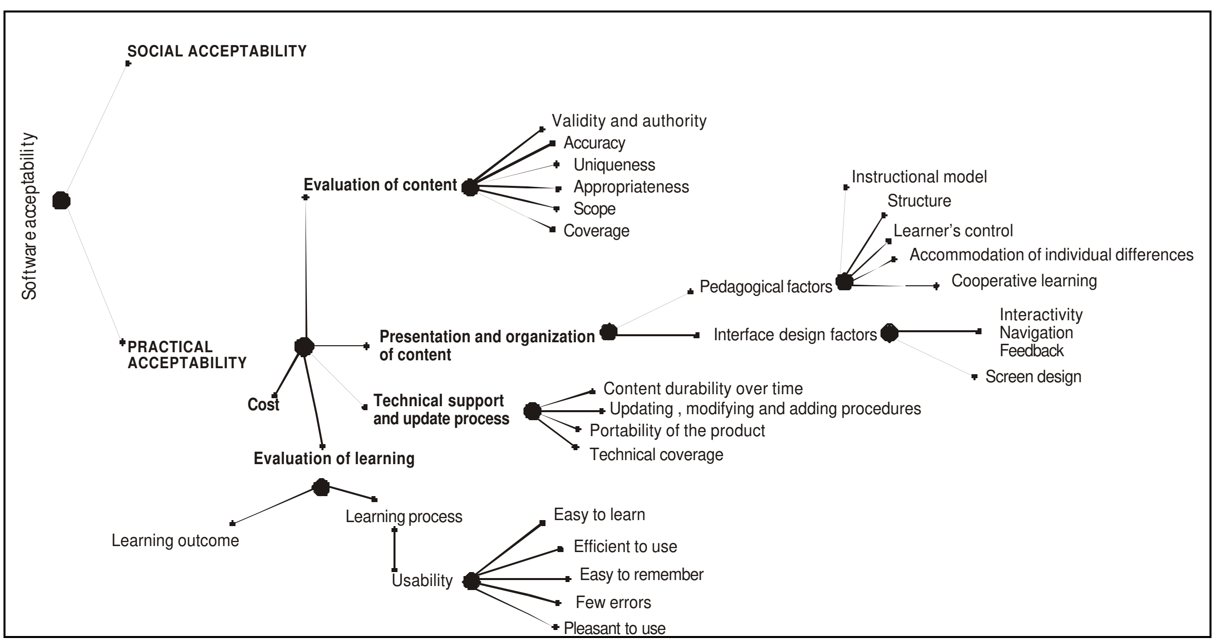

Figure 1. Diagram of the Evaluation Framework

All the above issues were used as a basis for the development of evaluation instruments mainly in the form of a suitability scale questionnaire and the evaluation was conducted from a team of researchers on educational technology in the University of Macedonia. The items included in the questionnaire were fixed alternative, with six-point scaled format from 'strongly disagree' (1) to 'strongly agree' (5) including a 'non-applicable' point (0). Some example items follow that are concerned with the Interactivity – Navigation – Feedback sector, included in Interface Design Factors (see Tessmer, Jonassen, & Caverly 1989).

1. The system has help key to get procedural information	**0 1 2 3 4 5**
2. The system has answer key for answering a question	**0 1 2 3 4 5**
3. The system has glossary key for seeing the definition of any term	**0 1 2 3 4 5**

Data Processing

The evaluation was conducted for each one of the titles mentioned above and then the sum of the scores of all items for every title was compared with the scores of the other titles. However, not only the overall sum was examined but also the sum for any given sector in practical acceptability i.e. Content, Presentation and Organization of the Content, Technical Support and Update Process and finally, the Evaluation of Learning. This approach was chosen in order to ensure the case when a title is pedagogically sound but lacks in terms of interface design and vice versa. The following table (Tab. 1) presents the results of the evaluation in percentage mode where 100% corresponds to the maximum value. Despite that every factor and sub-factors were examined separately during the evaluation process, the table presents principally the results of the main factors in order to allow easy interpretation. For example, the Interface Design Factors included in the 'Presentation and Organization of the Content' sector integrate Interactivity-Navigation-Feedback and Screen Design parameters as shown in (Fig. 1). However, (Tab.1) gives a single percentage for Interface Design. Also, it has to be mentioned that with regards to the evaluation of learning, Table 1 is concerned only with the Learning Process, i.e. Usability Evaluation and not with the Learning Outcomes which are typically evaluated with performance tests or assignments used to judge the quality and the quantity of learning as resulted scores are typically interval or ratio values (or can be transformed as such) so that powerful inferential statistical analysis can be employed to make generalizations about uniform impact.

		Percentage of social acceptability	Overall percentage of practical acceptability	Evaluation of each sector involved in Practical acceptability				
				Content	Presentation and Organization of the Content		Technical support & Update Process	Usability
					Pedagogical Factors	Interface Design Factors		
1	**An Introduction to MSW Logo**	80%	73%	86%	56%	66%	67%	90%
2	**How C Programming Works**	80%	70%	85%	62%	63%	50%	88%
3	**Introduction to C Programming I**	80%	79%	90%	72%	74%	70%	90%
4	**Introduction to C Programming II**	90%	67%	80%	65%	59%	51%	78%
5	**Logo**	90%	77%	86%	66%	73%	68%	92%
6	**LOGO Programming**	80%	65%	75%	62%	53%	63%	72%
7	**Pascal Programming v2.1**	80%	78%	93%	70%	72%	62%	92%
8	**rLogo**	80%	65%	86%	61%	55%	47%	78%

Table 1. Results of the Evaluation in a Percentage Mode

Discussion

This section is concerned with the conclusions derived from the processing of the data, during the evaluation of the software. It is obvious that all the software that has been evaluated has achieved satisfactory gradation in both Social Acceptance and Content Evaluation fields. On the contrary, the majority of the software presents the poorest performance on the 'Presentation and Organization of the Content' field that include Pedagogical and Interface Design factors and in turn the latter include screen design and interactivity-navigation-feedback parameters as shown in (Fig. 1). This difference shows that authors and producers of educational software often still focus on issues related with the validity and authority of the content in expense to issues related with interface design, despite the fact that hypermedia systems provide the necessary technology for highly interactive and potentially adaptive learning environments. Reigeluth and Curtis argued in 1987 that "the failure of so many instructional programs and materials has often been the result of an emphasis solely on content, with little regard for principles of instructional design to produce effective, efficient, and appealing instruction" (Reigeluth & Curtis 1987). The evaluation results show that their argument is still valid; fourteen years after.

Moreover, only four out of eight software examined contain exercises: How C Programming works, Introduction to C Programming I, Introduction to C Programming II, and Pascal Programming v2.1, and from these only two titles support interactive exercises with the appropriate feedback: Introduction to C Programming I and Pascal Programming v2.1. The other titles are limited strictly to simple examples that enhance the understanding but do not support collaboration or interactivity. In addition, it was also noticed that most of the examined software do not take into account the diversity that characterizes most of the students, like different previous experience in some learning fields, motivation, ability of assimilation, etc. With regards to Screen Design, all titles exhibit the information using basically text, images and graphics and not any other screen elements that could motivate the learner and assist him/her in retaining and recalling the information (i.e. video, animation etc.). Only Logo use some moving images as presentation means. Regarding the 'Technical Support' field almost half of the software has inadequate technical support from the designers or the corresponding company. Even fewer software companies offer additional instructions or suggestions for the teacher or even some kind of relative documentation on the potential of the software and teaching scenarios. Information about the hardware and software needed is often limited and inadequate. Additionally, we have concluded that the titles exhibit a lack in information concerning the last update or the frequency in which the site was renewed.

It is generally accepted that the Usability field has achieved relatively high scores in almost all of the cases. Usability is usually associated with five parameters (see Nielsen 1990): (1) Easy to learn: The user can quickly get some work done with the system, (2) Efficient to use: Once the user has learnt the system, a high level of productivity is possible, (3) Easy to remember: The casual user is able to return to using the system after some period without having to learn everything all over, (4) Few errors: Users do not make many errors during the use of the system or if they do so they can easily recover them, (5) Pleasant to use: Users are subjectively satisfied by using the system; they like it. All the environments examined were easy to use and the navigation procedures were simple in understanding even for the moderate student.

Finally, the best scores in all fields were exhibit by "Introduction to C Programming I" that is an in depth tutorial covering the C programming language. It assumes little or no knowledge of the C language, but does assume some knowledge of programming in general. It includes interactive multiple-choice exercises that can help the students to study and understand the issues presented.

However, as an overall conclusion we could say that despite the plethora of educational software available in the market the need for continuous research on evaluation methods and techniques is profound, as educational software has to be pedagogically and technically sound, in order to contribute meaningfully in the improvement of the learning experience. And above all designers and evaluators should always be aware that if educational hypermedia is not well designed, they will create psychological problems for users, such as memory overload and divided attention, or they will fail to suit the variety of ways that people work together or alone (see Preece 1993).

References

CEC. (2000). *e-Europe: An Information Society for all*. Brussels: EC;OOPEC.

Georgiadou E., & Economides A. Evaluation Factors of Educational Software. *IWALT 2000, International Workshop on Advanced Learning Technologies*, 2000, IEEE, New Zealand, Palmerston North. 113-116.

Nielsen, J. (1990). Evaluating Hypertext Usability. In D. Jonassen, & H. Mandl (Ed.), *Designing Hypermedia for Learning* (pp.147-168). Berlin Heidelberg New York: Springer-Verlag..

Tessmer, M., Jonassen, D., Caverly, D. (1989). *A Nonprogrammers Guide to Designing for Microcomputers*. Englewood Colorado: Libraries Unlimited Inc.

Preece J. (1993). Hypermedia, Multimedia and Human Factors. In C. Latchem., J. Williamson, & L. Henderson-Lancett (Ed.), *Interactive Multimedia: Practice and* Promise (pp.135-149). London: Kogan Page..

Reigeluth C.M., & Curtis, R.V. (1987). Learning Situations and Instructional Models. In R. Gagne, (Ed.), *Instructional Technology:* Foundations (pp.175-206). Hilsdale, NJ: LEA.