Georgiadou, E., Economides, A.A., Michailidou, A. & Mosha, A.: **Evaluation of educational software designed for the purpose of teaching programming.** *Proceedings 9th ICCE SchoolNet 2001 International Conference on Computers in Education*, pp. 745-752, AACE 2001.

Evaluation of Educational Software Designed for the Purpose of Teaching Programming

Dr. Elissavet Georgiadou , Dr Anastasios Economides, Anna Michailidou, Anna Mosha
University of Macedonia, Greece
<elisag@otenet.gr>, <economid@uom.gr>, <michan@uom.gr>, <mosha@uom.gr>

Abstract

In this paper we evaluate popular educational software designed for the purpose of teaching programming available in the market that addresses mainly beginners and intermediate students. The evaluation is conducted on the basis of an evaluation framework built in the University of Macedonia, Greece in the context of project "EPENDISI" that is mainly designed to introduce computer and network technologies into the everyday school practice, especially in secondary education, and also to build a database that contains information and resources on several evaluated educational software on almost all secondary school subjects. The paper first presents the educational software that was evaluated and then it discusses the evaluation framework. Next, it describes the evaluation method and the processing of the data and finally reports some conclusions derived during the evaluation of the educational software.

Keywords: Educational software evaluation, Criteria of educational software evaluation

1. 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'. 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 are associated with this reform. One of these is the Project EPENDISI (www.ithaca.uom.gr) run by the University of Macedonia that aims to train secondary schoolteachers in the use of ICTs in the classroom and also to build a database that contains information and resources on several evaluated educational software on almost all secondary school subjects. The pieces of educational software that were evaluated were selected on the basis of their usefulness and completeness in accordance with the school curriculum. The evaluation of the software is conducted on the basis of an evaluation framework built in University of Macedonia, that examines both social and practical acceptability of educational software. This paper will present the evaluation results on fourteen educational software designed for the purpose of teaching programming that addresses mainly beginners and intermediate students. The following section describes briefly all fourteen educational software titles that were evaluated.

2. EDUCATIONAL SOFTWARE PRESENTATION

1. **Beginners Basic Helpfile** (www.pc-developer.com/qbasic/register.htm) is a tutorial for people who would like to learn QBasic programming. It introduces Qbasic through eight lessons, three programming exercises, and basic technical articles. The lessons aim to help users to create their first programs. Programming code is highlighted for users to test and try out, as the text explains exactly what needs to be done and why. Technical tips are provided for extra insight. Registration provides 12 additional lessons, supplemental articles, an email newsletter, tips and a support pack with Qbasic programs to load and test.

2. **CoffeeCup HTML Express, Version 5.0** (www.davecentral.com/12914.html) walks users through each step of the creation of web pages. Company suggest that is perfect for beginners who want to get their first web page up and running fast, but complex enough for professionals who need a fast start on a project.
3. **Courseware Web-HTML** (downloads.mediadna.zdnet.com/info/com.zdnet_downloads_000VK8_ 000VK8.html) is a small tutorial that explains how to create Web pages using Notepad and HTML. It introduces customary tags and walks users through the process of adding and manipulating text and graphics, linking pages, create ordered lists, and more. The course uses notated examples, which explain exactly how the code is used. An assignment is provided to give users hands-on experience with creating their first Web page.
4. **Hypertext Guide** (www.spriteworks.com/hyper.html) is a tutorial that explains how to create a Web page using HTML, ActiveX, Dynamic HTML, and cascading style sheets. Every topic includes a host of source-code examples and tables of the various tags, including explanations of what each tag does and why. Tips are found throughout the illustrated guide, complete with working examples. Registration provides additional topics, including Forms, Tables, and Multimedia in addition to hundreds of icons, textured buttons, and more.
5. **L-Basic** (www.cyber-matrix.com/lbasic.htm) is a Visual Basic language tutorial that focus strictly on writing code. Topics include variables, conditional statements, string functions, code design and more. Because the tutorial does not focus on the visual aspects of programming, there are no graphics included. Instead, code techniques are demonstrated through the use of Message Boxes.
6. **Liberty BASIC** Course Materials (world.std.com/~carlg/mtls.html) is a series of mini-tutorials designed to be completed over a six-week period for the Liberty Basic programming language. Liberty BASIC is a Windows programming tool that brings BASIC's ease of use to Microsoft Windows. It includes a BASIC language for Windows, a Visual Development Tool, Freeform, an editor for writing BASIC programs, an easy to use tracing debugger, easy calling of DLLs and APIs and a programmable spreadsheet.
7. **LogoMation** (www.magicsquare.com/LM2/) is a computer-programming environment for Middle and high-School students, learning computer programming, which is based on the principal of Logo. The user write a program that describes a picture, either still or animated, to the computer. When s/he clicks GO, it draws what was described. While the commands are fairly simple, programs get complex quickly. The included LogoMation book provides step-by-step instructions on using the development environment.
8. **Mach Turtles Logo Learning Edition** (www.machturtles.com/download.htm) is a version of Logo that provides multimedia capabilities including animation, sound playback, MIDI sound output, and full-color graphics support. Mach Turtles Logo Learning Edition is free for personal and educational use.
9. **Microprocessor Simulator** (download.cnet.com/downloads/0-10031-100-891195.html?tag=st.cn.sr1.dir) is for students who are learning low-level programming for the first time. After learning the basics with the simulator, it is much easier to understand what a real-life CPU and assembler do. In the unregistered version of the simulator, arithmetic, logic, input/output, and jump instructions are available. In the registered version, users can code procedures and interrupts and use CALL, RET, INT and IRET commands. Moreover, recursive procedures and functions are possible.
10. **MicroWorlds Pro** (www.lcsi.ca/) allows students to create dynamic, interactive school and Internet projects using Logo. It lets students become active web designers not just passive web viewers. Teachers and students can use MicroWorlds Pro to enhance their understanding of MicroWorlds and to get a real sense of the depth and breadth of this multimedia-programming environment.
11. **Pascal Programming Tutorial 1.0** (download.cnet.com/downloads/0-10033-100- 904799.html?tag=st.cn.sr1.dir) is a tutorial specifically designed for use in schools and colleges. It uses working examples to illustrate the core features of the language; starting at a very basic level, but progressing to more advanced concepts. The topics covered include program design, numeric and text variables, branching, looping, procedures, arrays, functions, parameters, records, file processing, and recursion.
12. **Visual Basic Introduction** (for VB 5/6) (downloads.mediadna.zdnet.com/info/com.zdnet_downloads_0010I4_0010I4.html) is an introduction to programming in Visual Basic. The shareware version includes about a quarter of the material in the registered version, taking the user on a tour of the VB programming interface, through building his/her first project, to basic branching constructs and user interaction. The full course extends this coverage to include topics such as conditional loops, check boxes and radio buttons, the timer control, menus, dialog boxes, arrays, text handling, structured data types and sorting.
13. **Web Resources' Tutorials** (www.nr1webresource.com/tpack/index.htm) includes more than thirty original tutorials covering major aspects of Web design, such as JavaScript, HTML, CGI. The tutorials are on advanced subjects but are written in a way that beginners and intermediate students of Web design can understand. In version 6.0, the package was completely rebuilt to feature the best tutorials from all over the Web in one package. Version 6.4 adds a few tutorials and makes layout improvements.

14. **WinHTML** (www.brokersys.com/gcsoftware/) helps users to learn HTML. It includes features for creating, editing, and maintaining web pages. It also includes support for HTML forms and tables, List Assist for easy creation of lists, and some of the more popular HTML markups and Netscape extensions.

3. EVALUATION FRAMEWORK OF EDUCATIONAL SOFTWARE

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 processes 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. Figure 1 presents all factors included in the framework in a diagram, and follows a more detailed account of them. However, due to the lack of space the framework is presented here in a short version, however, all the details about it can be found in "Evaluation Factors of Educational Software" (Georgiadou & Economides, 2000).

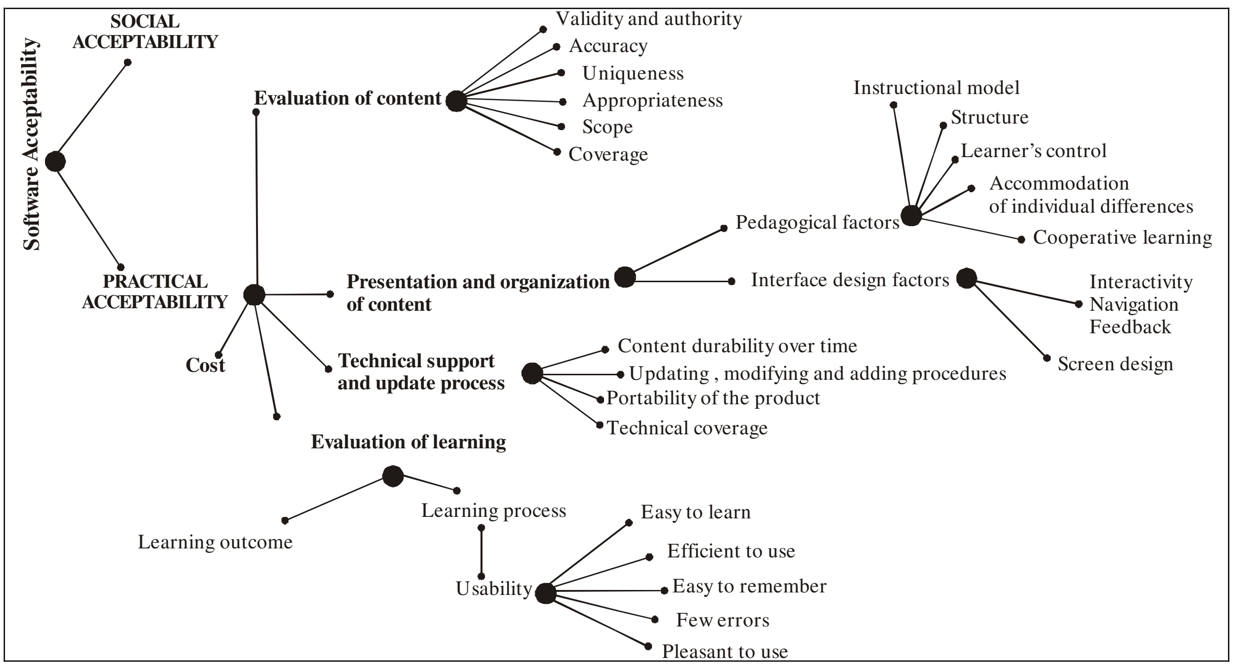

Figure 1: Diagram of the Evaluation Framework

3.1. CONTENT

The information contained within a piece of educational software is the first parameter that should be evaluated according to the following criteria (Netskills, 2000; NCPD, 2000):

- Validity and Authority: Reliable content; Reputable authors; publishers and origin of information.
- Accuracy: Current and error-free information; Bias-free viewpoints and images; Correct use of grammar.

– Appropriateness: Concepts and vocabulary relevant to learners' abilities; Information relevant to age group curriculum; Interaction compatible with the physical and intellectual maturity of intended audience.
– Scope and Coverage: Information of sufficient scope and depth; Logical progression of topics; Variety of activities with options for increasing complexity.

3.2. PRESENTATION AND ORGANIZATION OF THE CONTENT

The factors associated with this sector are the pedagogical ones that are concerned with learning and instructional design theories and the interface design factor.

Pedagogical Factor. This is a complicated factor as there are different beliefs on how humans learn. However, cognitive theories stress that learning is an active, constructive, cumulative, self-regulated process in which the learner plays a critical role. Moreover, current instructional theory focusing on constructivist approaches depends on information access and learning environments that encourage free interaction with information. The agreement with the principles of an instructional design theory depends heavily on the subject matter. In addition, teacher's belief is of great importance, especially in cases when the educational software is part of the curriculum. Reigeluth and Squire suggest the integration of instructional design theories in the form of an 'umbrella theory' in order to decide when to use a particular one (1998).

Nevertheless, the two core elements that are important in all educational settings are 'motivation' and 'structure', which largely define the instructional nature of an information environment. A typical way to motivate the learner is to inform him/her what will s/he achieve at the end of the instruction by stating the aims and objectives (Gagne et al, 1988). As far as the structure is concerned, that is how to organise instructional information, again depends on the subject matter. However, in cases when the instructor wants to permit the learners to advance, review, see examples, repeat the unit, or escape to explore another unit, Jonassen suggest the network structure as most appropriate, which implies an explicit organisation or arrangement of nodes and associative links (Jonassen, 1992).

In hypermedia learning systems another important element is 'learner's control', which is primary in the design of interactive learning as it allows students to tailor the learning experience in their own individual needs. However, there are dangers in surrendering too much control to the user. Low-ability students may get confused when control depends on a wide range of options (Gray, 1989; Litchfield 1993). The high level of learner control may result in disorientation and distraction. The amount and type of learner control depends on the learner characteristics (age and cognitive capabilities), content, and the nature of the learning task (Poncelet & Proctor 1993). Content that must be mastered and unfamiliar tasks often requires more program control, compared to content with no qualified mastery levels or familiar learning tasks. Learner control is more appropriate than program control when learners are more capable and are familiar with the learning task. Moreover, advisement is provided to assist them in making decisions and control is used consistently within a lesson (Ross & Morrison 1989). In general, the more control is given to the learners, the more feedback about their decisions should be given (Mcateer & Shaw, 1995).

Moreover, the issues of 'accommodation of individual differences', and 'cooperative learning' are highly important in the effectiveness of hypermedia-based learning. In most education contexts learners are not homogeneous in terms of prerequisite knowledge, motivation, experience and learning styles. Also evidence suggests that when hypermedia learning systems are structured to allow cooperation, learners benefit both instructionally and socially.

Interface Design Factor. *Interactivity - Navigation - Feedback*: Interactivity in instruction comprises the nature of the activity performed by the technology and the learner, as well as the ability of the technology to adapt the events of instruction in order to make that interaction more meaningful (Reigeluth, 1987). It is important to design as much meaningful interactivity as possible into instructional software. Guidelines for increasing interactivity in instructional programs according to Orr, Golas and Yao (1994) are: Provide opportunities for interaction at least every three or four screens; Chunk the content into small segments and build in questions, reviews, and summaries for each segment; Ask students to apply what they have learnt rather than memorize. Also, use rhetorical questions during instruction to get students to think the content and consider designs where the learner discovers information through active exploration.

The amount of navigational assistance needed is a function of the size of the knowledge base, the usefulness of navigational aids that are already part of the authoring software, and the types of links the software allows (Locatis et al, 1989). According to Shneiderman (1989) navigation may be facilitated when users can: Back up a node at a time, Review their paths and immediately re-access any previous node; Search for information with key words and use maps and tables of contents to see the overall structure of the knowledge base; Get 'fish eye' views indicating the names or contents of nodes neighboring the one currently displayed. A courseware should promote interactivity by assisting access to some or all of the following options (Tessmer et al, 1989): Help key to get procedural information; Answer key for answering a question; Glossary key for seeing the definition of any term; Objective key for reviewing the course objectives; Content map key for seeing a list of learner commands or options available and overview of introduction key for reviewing the introduction to the unit; Menu key for exiting the lesson and returning to the menu and exit key for exiting the course. Also, summary and review key for reviewing whole or parts of the lesson; Comment key for recording a learner's comment; Examples key for seeing examples of an idea and finally keys for moving forward or backward in a lesson and for accessing the next lesson in a sequence.

The basic factors that can determine the effectiveness of feedback are the type and frequency of feedback given and the delay between feedback and instruction (Jonassen & Hannum 1987). Feedback is closely related with the issue of interaction, as action without feedback is completely unproductive for a learner (Laurillard, 1993). Many actions require more extended extrinsic feedback than confirmation feedback. Simple answers such as right or wrong cannot provide information about how learners should correct their performance. On the contrary, correct response and explanation feedback would give the learners information about how to adapt and correct their performance. Some basic guidelines on performance feedback are (Orr et al, 1994; Jonassen & Hannum 1987): Provide feedback immediately after a response; Vary the placement of feedback according to the level of objectives. Provide feedback after each response for the learning of lower level objectives, and at the end of the session for the higher level ones; Provide feedback to verify the correctness. For incorrect responses, give the student information about how to correct their responses, or hints to try again; If possible, allow students to print out their feedback.

‘Screen design’ is also an important evaluation factor. Different screen elements should be used to present stimulating information that will motivate and assist the learners in retaining and recalling the information. The psychological limitations to consider when designing hypermedia learning systems include: (a) Memory load: i.e. how many different control icons is it reasonable for learners to remember at any one time? (b) Perception: i.e. what colors and fonts provide the best readability?, and (c) Attention: i.e. how can the users' attention be drawn to information that is relevant, when there is a lot of different information on the screen? (Preece, 1993). Researchers have produced screen design guidelines that are concerned with use of space, colors, fonts, graphics, video, animation etc. however due to lack of space they are not included in the paper. A good summary of these issues can be found in “Principles of Screen Design for Computer Based Learning Materials”, written by A. Clarke (1992).

3.3. TECHNICAL SUPPORT AND UPDATE PROCESS

It is a common belief that new knowledge comes to life every day and therefore educational instruments should be regularly updated. The rapid growth of technological innovations is also important, as technology is the delivery platform of educational software. Therefore the value of the information content and the product over time should be examined. Some issues to consider are (Netskills, 2000): a) Durability of the content over time, b) Updating, modifying and adding procedures, and c) Portability and technical coverage of the product. When the educational software is web-based, then system and site integrity should be evaluated as well by examining the stability of links, mirror sites, the adequacy of administration and maintenance, the regular updating and finally the availability of archived information.

3.4. EVALUATION OF LEARNING

Marchionini (1990) argues that the interactivity of hypermedia systems provides learners with access to vast amount of information in varied forms, control over the process of learning, and the potential for collaboration with the system and other people. Such empowerment of learners forces evaluators of learning to adopt a broad-based set of methods and criteria to accommodate 'self-directed' learning. He proposes a 'multi-faceted' approach to the evaluation of hypermedia based learning that address both the outcomes and the processes of learning. The learning outcomes are evaluated through performance tests typically used to judge the quality and the quantity of learning, which usually have the form of ‘pre-tests’ used to determine learning outcomes prior to the intervention and ‘immediate’ and ‘delayed post-tests’ to examine learning outcomes after the intervention. The learning process refers to the usability of a product and should be evaluated by observing and measuring the end-users attitudes. Usability is usually associated with five parameters (Nielsen, 1990): (1) Easy to learn: Users 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.

4. EVALUATION METHOD

All the above issues were used as a basis for the development of evaluation instruments mainly in the form of a suitability scale questionnaire. 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). The evaluation was conducted for all the titles from project researchers specialize in educational technology. 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. Table 1 presents the results of the evaluation in a percentage mode where 100% corresponds to the maximum value.

		Percentage of social acceptability	Overall percentage of practical acceptability	Evaluation of each sector involved in Practical acceptability				
				Content	Presentation and Organisation of the Content		Technical support and Update Processes	Usability
					Pedagogical Factors	Interface Design Factors		
1	**Beginners Basic Helpfile**	90%	67%	87%	68%	59%	60%	75%
2	**CoffeeCup HTML Express, Version 5.0**	90%	62%	86%	61%	55%	50%	70%
3	**Courseware Web-HTML**	70%	62%	78%	61%	55%	52%	77%
4	**Hypertext Guide**	70%	60%	70%	62%	58%	40%	65%
5	**L-Basic**	70%	59%	73%	52%	56%	50%	82%
6	**Liberty BASIC Course Materials**	80%	66%	78%	60%	68%	57%	75%
7	**LogoMation**	90%	60%	80%	58%	50%	66%	67%
8	**Mach Turtles Logo Learning Edition**	80%	62%	85%	58%	54%	69%	70%
9	**Microprocessor Simulator**	90%	65%	77%	62%	60%	75%	65%
10	**MicroWorlds Pro**	90%	81%	91%	70%	82%	100%	80%
11	**Pascal Programming Tutorial 1.0**	80%	67%	90%	59%	62%	52%	82%
12	**Visual Basic Introduction**	80%	63%	76%	62%	58%	48%	80%
13	**Web Resources' Tutorials**	90%	67%	85%	58%	65%	60%	80%
14	**WinHTML**	80%	61%	77%	52%	57%	71%	65%

Table 1. Results of the Evaluation in a Percentage Mode

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 processes 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. 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 of the results. 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 Figure 1. However, Table 1 gives a single percentage for Interface Design that is the average of the above parameters. 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.

5. DISCUSSION

In this section we discuss the evaluation results and derive some conclusions. It is obvious that all the evaluated software has achieved satisfactory grades in both the Social Acceptability and the Content Evaluation fields. In the contrary, the majority of them presents the poorest performance in the Presentation and Organization of the Content sector. This difference shows that authors and producers still focus mostly on the validity and appropriateness of the content and not equally in ways that content is presented and organized, despite the fact that hypermedia systems provided the necessary technology for highly interactive and potentially adaptive learning environments. Reigeluth and Curtis argued in 1987 "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 today, fourteen years after.

Most of the titles present the information using basically text and graphics and not any other screen elements that could motivate learners and assist them in retaining and recalling the information (i.e. video, animation etc.). In only very few software packages, there was a successful combination of sound and video along with the most suitable means of presentation. Moreover, very few of the evaluated software contain interactive exercises with the appropriate feedback. Most of them are limited strictly to simple examples that enhance understanding but do not support collaboration or interactivity. In addition they do not take into account the diversity that characterizes most of the students in learning experience, motivation, ability of assimilation, etc.

Furthermore, 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 concerning the potential of the software as well as teaching scenarios. Information about the required hardware and software is often limited and inadequate. It is noticed that generally the field of Evaluation of Learning, which as explained in section 4 Evaluation Method refers to the usability of the products, has achieved relatively high scores in almost all of the cases. The environments were easy to use and the navigation procedures were simple to be understood even for the moderate student. Finally, the best scores in all fields were exhibit by the "MicroWorlds Pro" software that allows students to create dynamic, interactive Internet projects using Logo. The tutorial is fully integrated with the Web permitting students to load their favourite interactive projects onto the Web and allowing teachers to locate curriculum ideas in the Library section of product's web site. MicroWorlds Pro has been presented with the "2000 Districts' Choice Top 100 Products" award from Curriculum Administrator magazine, a fact that validates in a sense the evaluation results. To select this year's top 100 products, Curriculum Administrator tabulated reader responses, interviewed school leaders, and drew on the collective experience of their editorial staff to determine which products were truly enhancing the work of teaching and learning.

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" (Preece, 1993).

6. REFERENCES

Clarke, A. (1992) *Principles of Screen Design for Computer Based Learning Materials*. U.K: Department of Employment.

Gagne, R., Briggs, L., Wager, W. (1988) *Principles of Instructional Design*. N.Y: Holt, Rinehart & Winston.

Georgiadou E., and Economides A. (2000) "Evaluation Factors of Educational Software", in Proceedings IWALT 2000, International Workshop on Advanced Learning Technologies (New Zealand, Palmerston North, 4-6/12/2000), IEEE, 113-116.

Gray S.H. (1989) "The effect of locus of control and sequence control on computerised information retrieval and retention", *Journal of Educational Computer Research* 5.4, 459-471.

Jonassen, D.H., Hannum, W. (1987) "Principles for Designing Computer Software", *Ed. Tec*., 1.18, 42-51.

Jonassen, D.H. (1992) "Designing Hypertext for Learning". In Scanlon, E., O'Shea, T. (eds.): *New Directions in Educational Technology*. Berlin: Springer-Verlag, 123-130.

Laurillard, D. (1993) *Rethinking University Teaching.* London: Routledge.

Litchfield, B. (1993) "Design Factors in Multimedia Environments: Research Findings and Implications for Instructional Design", *Annual Meeting of the American Educational Research Assossiation*, 1-10.

Locatis C., Letourneau G., and Banvard R., (1989) "Hypermedia and Instruction", *Educational Technology Research and Development*, 37.4, 65-77.

Marchionini, G. (1990) "Evaluating Hypermedia Based Learning". In Jonassen D., Mandl H. (eds.): *Designing Hypermedia for Learning*. Berlin: Springer-Verlag, 355-374.

Mcateer, E., Shaw, R. (1995) *The Design of Multimedia Learning Programs*, Univ. of Glasgow: EMASHE Group.

Netskills, (2000) *Internet Detective*. www.netskills.ac.uk/TonicNG/content/detective/.

Nielsen, J. (1990) "Evaluating Hypertext Usability". In Jonassen, D., Mandl, H. (eds.): Designing Hypermedia for Learning. Berlin: Springer-Verlag, , 147-168.

NCDP (North Carolina Department of Public Instruction), (2000) *Criteria for Evaluating CD-ROM*, ww.sret.sreb.org/criteria/

Orr, K. L, Golas K. C., and Yao K. (1994) "Storyboard Development for Interactive Multimedia Training", *Journal of Interactive Instruction Development*, Winter, 18-31.

Poncelet G.M., Proctor L.F. (1993) "Design and Development Factors in the Production of Hypermedia-based Courseware", *Canadian Journal of Educational Computing* 22.2, 91-111.

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

Reigeluth C. M. (1987) *Instructional Theories in Action,* Hilsdale New Jersey: Lawrence Erlbaum Associates.

Reigeluth C.M., Curtis, R.V. (1987) "Learning Situations and Instructional Models". In Gagne, R. (ed.): *Instructional Technology: Foundations*. N.J: LEA, 175-206.

Reigeluth, C., Squire, K. (1998) "Emerging Work on the New Para- digm of Instructional Theories", *Educational .Technology* 88.4, 41-47.

Ross S.M., Morrison G.R. (1989) "In Search of a Happy Medium in Instructional Technology Research", *Educational Technology Research and Development,* 37.1,19-33.

Shneiderman, B., Kearsley, G. (1989) *Hypertext hands on*!, Reading Massachusetts: Addison Wesley.

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