In: Technology and Teaching
Editors: Jeff Sigafoos and Vanessa Green, pp.
ISBN 978-1-60021-699-2

Chapter 11

EVALUATION OF EDUCATIONAL SOFTWARE: THEORY INTO PRACTICE

Quynh Lê and Thao Lê

ABSTRACT

Educational software has proliferated in virtually every academic discipline and at various levels from preschool to postgraduate studies. With the rapid development of computer technology, educational software has been greatly improved to cater for a wide range of teaching and learning experiences. This chapter discusses different ways in which educational software is evaluated. Teachers firmly hold certain views on learning and teaching which reinforce their adoption of certain evaluation criteria. As educational software reflects how teaching and learning is viewed, educational metaphors are used as an evaluative approach to examine the theory and practice of software evaluation in education and for education.

Key words: educational software, evaluations, educational metaphors.

INTRODUCTION

Computer or information technology has permeated the field of education at various levels. It has created a big challenge to traditional teaching. While some educators are reluctant to take the challenge, others make use of computer technology to enhance teaching and learning. Educational software can play an important role in enhancing the educational discourse. However, like all products on the market, not all educational software is suitable and productive for teaching and learning. Thus awareness of issues related to the evaluation of educational software is important to educators. This chapter examines different ways in which educational software is evaluated.

Most criteria used to evaluate educational software reflect strongly the principles of teaching and learning adopted by evaluators. They are the rationales which drive the construction of software. To capture this dynamic connection between software and educational rationales in software evaluation, metaphor is used an as evaluative approach to highlight the theory and practice of software evaluation as they symbolize different roles that software plays in teaching and learning.

DEFINING EVALUATION

There are many definitions of the term evaluation. The common agreement is that evaluation is about assessment of quality of a product, task, program, or activity. Evaluation is one of the most important aspects of teaching and learning. If it is well planned and conducted, evaluation gives many valuable insights about aims, achievement, learners' attitudes and learning styles, teachers' performance, learning environment, and course improvement.

According to Wilss (1997), productive learning is dependent upon considerations such as content, delivery, the needs of the learner, and specific aspects of the learning environment. Evaluation of courses that incorporate technological innovations help to illuminate factors that may contribute to educationally sound practices and productive learning outcomes.

Traditionally, evaluation is conducted primarily to find out about students' ability to undertake a particular task or course. The results gained from this kind of evaluation are often used to reinforce evaluators' expectations and assumptions. Bain and McNaught (1996) point out that teachers firmly hold certain views on learning and teaching which reinforce the adoption of their teaching approaches.

Teaching styles and the use of educational resources reflect the views of teachers on how learners 'should' learn. The choice of educational software is also determined by the educational principles that they adopt. As the amount and variety of educational software grows, there is a commensurate need for it to be assessed for suitability for its intended purpose. Teachers need to know whether and how an item can be used to improve their teaching; learners need to know how it might impact the learning experience (Hammond, Trapp, & McKendree, 1994).

As computers have become more rapid and powerful, educational software has flourished and there are numerous claims made by software developers in order to sell their products to teachers. Thus, evaluation of educational software is important so that teachers can make an appropriate choice of the software which reflects their educational principles and which is appropriate to the teaching and learning context. While numerous software packages saturate the educational market, there is a growing number of evaluative criteria and checklists (for examples CEISMC, 2006). Instead of adding to the already large number of checklists for software evaluation, we propose a metaphor approach to address evaluation of educational software. The rationale and implementation of this approach will be presented in the rest of this chapter.

METAPHOR AS AN EVALUATIVE APPROACH

According to Henkel (2006), great emphasis is being placed on understanding teacher's thought patterns, or the reflection that occurs about the teaching-learning process. Thinking about this process in different ways is crucial for teachers to most effectively understand and carry out their roles in the classroom and school more broadly. Teachers' roles are most effectively carried out if they are consistent with a coherent educational philosophy. Metaphor concepts present in teachers' communication with colleagues, students and their parents give meaning to teachers' views about teaching and learning.

We may not always know it, but we think in metaphor. A large proportion of our most commonplace thoughts make use of an extensive, but unconscious, system of metaphorical concepts, that is, concepts from a typically concrete realm of thought that are used to comprehend another, completely different domain. Such concepts are often reflected in everyday language, but their most dramatic effect comes in ordinary reasoning (Lakoff, 1995).

Metaphors are constantly used by teachers when they talk about their teaching, children's learning, and most importantly their views, assumptions, and beliefs about teaching and learning. Metaphors are used consciously by teachers to draw attention to important concepts and ideas in education. However in most cases, metaphors have become so widely used in an educational discourse that its metaphorical nature may not be noticed. The following sentences occur very often in teachers' communication.

- Some children are like a blank sheet and we need to fill in for them.
- School is not the real world. It is a prison for some students and a home for others.
- Teachers and students are members of a learning community.
- The Computer is a tool. It is not a teacher.
- The Internet is an educational superhighway.

Metaphors may be used as a powerful tools for determining and expressing one's educational philosophy. Often other professionals and lay people are more receptive to ideas when they are expressed indirectly through symbolism. Metaphors use symbolism to link ideas about teaching and learning to something more familiar (Henkel, 2006).

The use of educational metaphors in evaluation of education software is illuminative as they are powerful in conceptual construction. Metaphor may create realities for us, especially social realities as they not only make our thoughts more vivid and interesting but they actually structure our perceptions and understanding (Lakoff & Johnson, 1980).

Instead of using traditional features for evaluating software such as screen design, navigation, text type, cost and user-friendliness, we use the concept 'metaphor' as a basis on which evaluation of software is conducted. By doing this, we assign significance to the educational aspect of education in software evaluation. The following metaphors are chosen as they represent major theoretical perspectives and software developments in educational multimedia:

- Software as a tool;

- Software as an instructor;
- Software as a facilitator of learning;
- Software as a virtual class.

The discussion of each metaphor starts with a brief presentation of the relevant theoretical perspectives and later examines how the metaphors underpin software evaluation. As previously stated, instead of adding to the number of checklists for software evaluation, we propose a metaphor approach to address evaluation of educational software. The points or questions given under each metaphor category should be used as illustrative items of evaluative criteria in the proposed metaphor approach to software evaluation.

SOFTWARE AS A TOOL

There are many types of software and they are produced and used for a purpose. In education, the most versatile function of software is instrumental. It is used for performing a certain function such as calculating, drawing, editing, proofreading, communicating, etc. Word processing is the most widely used software in education. Students use them to write assignments, produce documents, and communicate with others. Software for proofreading includes spell checking and grammar checking. Nowadays, most software is multifunctional. For example, word processing software can perform many complex functions, such as text editing, statistical analysis, communication and design.

Apart from the versatile software programs for general purposes such as WORD, EXCEL, ACCESS, PowerPoint and Publisher, some software programs are designed for specific purposes such as SPSS, STRATA and SAS for statistical data analysis, NVivo and Atlas-TI for qualitative data analysis, and EndNote for referencing.

Evaluation is often made in the form of a checklist used by developers, teachers and learners. A checklist is a list of items, which covers important points to be examined. According to Squires and McDougall (1994), many lists of criteria for assessment of individual packages have been used. They vary in content, length and style, but all have been designed in an attempt to help teachers choose software of educational value. Evaluation of software as a tool takes into account the following factors:

- The educational background of the target learners;
- Their Information and Communication Technology (ICT) literacy awareness;
- The user-friendly features;
- ‘Help’ facility;
- Cost; and
- Hardware support.

SOFTWARE AS AN INSTRUCTOR

This metaphor represents the instructionist view of teaching and learning. When software is used as an instructor, its primary role is to teach learners to develop knowledge and skills.

Normally it is hierarchically structured in terms of content and skill complexity. Each lesson focuses on a specific content or skill and all the lessons are arranged from introductory to advanced levels. Instruction is sequential in the sense that learners are expected to move sequentially from the early lessons to the final ones. Skipping lessons is discouraged as it interrupts the continuity of the structured instruction and could affect learners' progress and achievement. Drill-and-practice is a common feature of instructional software as it reflects strongly the Initial-Response-Evaluation (IRE) model, which consists of three sequential stages:

1. Initial stage: Instructors introduce the content in terms of definition, description and explanation.
2. Response stage: After having been exposed to instructions, learners are expected to gain knowledge or skills and know how to translate new knowledge or skills into practice.
3. Evaluation: Teachers evaluate learning through responses given by learners.

In the IRE model, learning objectives must be established at the beginning before a software item is developed. They are used to evaluate learning. Objectives used in education, whether they are called learning objectives, behavioural objectives, instructional objectives, or performance objectives are terms that refer to descriptions of observable student behaviour or performance that are used to make judgments about learning - the ultimate aim of all teaching (Kizlik, 2006).

The following geometry software prototype illustrates the metaphor of software as an instructor.

a) Lesson 1: Scale drawing
   i. What is scale drawing?
   ii. Some examples of scale drawing
   iii. Why do we need scale drawing?
   iv. Scale drawing exercises
b) Lesson 2: Parallelograms
   i. What is a parallelogram?
   ii. Three special types of parallelograms
   iii. Rhombus
   iv. Rectangle
   v. Square
   vi. Angles of a parallelogram
c) Investigation exercises

In summary, evaluation of educational software from the instructionist perspective takes into account the following factors:

- Does the software have clearly stated learning objectives?
- How are the learning objectives projected in the software?
- Is the software structured in terms of complexity and its developmental stages?

- Does the software provide items for drill-and-practice exercises?
- Does the software provide items for testing?

SOFTWARE AS A FACILITATOR OF LEARNING

This metaphor represents constructivism. Papert (1993) questioned the traditional view underlying educational software. His view is that educational software based on instructionism does not prepare children for the computer age. While the instructionist model of educational software focuses on the significance of instruction in teaching and learning, the constructivist model of educational software plays less attention to instruction and more on the active role of learners in the learning process. Learning is viewed as a meaning making process as learners bring their knowledge, experiences and world view to learning. In many aspects, instruction software is rather consistent in its presentation of form and content to learners; whereas constructionist software varies a great deal, not so much in terms of learning principles but more on the learning experiences and the kind of communicative interaction between learners and their teachers.

Turtle Geometry is a pioneer of constructionist educational software under the leadership of Papert. He is one of the pioneers of Artificial Intelligence and Logo at Massachusetts Institute of Technology (MIT). He is a mathematician, computer scientist and prominent educator. Logo is a programming language developed for children. It provides onscreen objects, widely known as Turtles. Children can direct Turtles to move on various paths by giving them commands. Thus they can create various shapes.

In language education, a constructionist prototype of web-based software typically consists of the following features:

- Topic introduction,
- Inquiry-based learning,
- Task-based learning,
- Case study as a basis for discussion and reflection,
- Questions and discussion,
- Illustrations,
- Suggested projects,
- Resources.

In summary, evaluation of educational software from the facilitator perspective asks the following questions which underlie the constructionist perspective.

- Does the software promote curiosity and inquiry?
- Does the software give choices for learners to control their own learning?
- Does the software present tasks for problem solving?
- Does the software provide collaborative learning experiences?
- Does the software provide interaction with others?
- Does the software provide educational tools and relevant resources for learners and teachers?

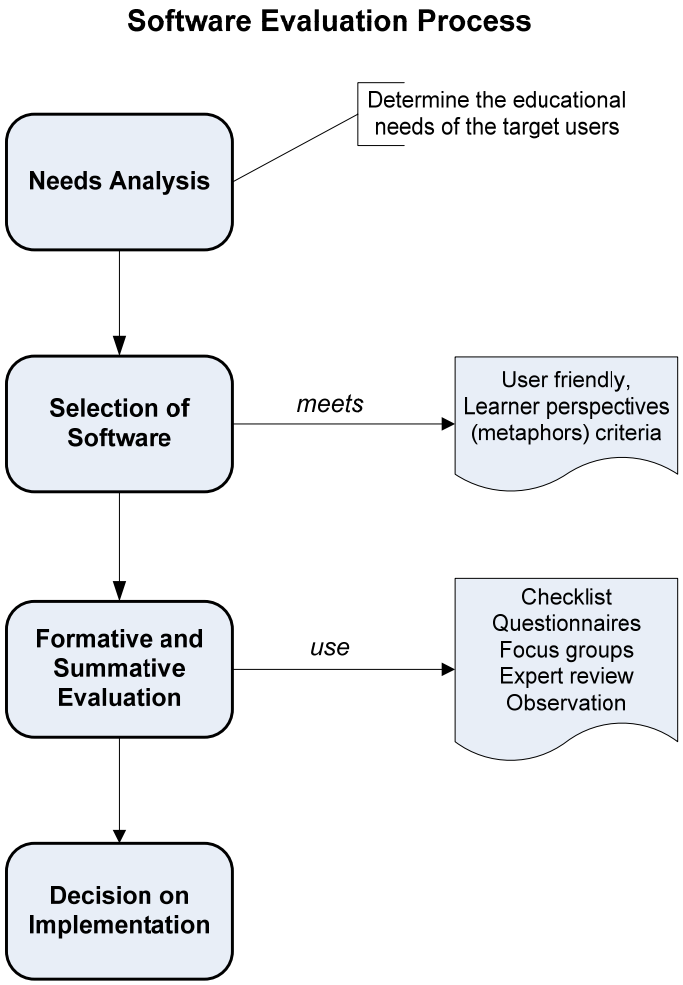
SOFTWARE AS A VIRTUAL CLASS

When we think of educational software, we tend to focus on small items of software such as programs teaching pronunciation, multiplication, grammar, and typing. These software programs are designed to teach a specific skill. There are also educational software packages which are designed for a targeted group of learners such as a web-based academic course and a multimedia-based training program. This type of software is called 'courseware'. For example, MIT's OpenCourseware (MIT, 2002-2006) is produced by MIT and it consists of a number of courseware for different courses such as architecture, health sciences and technology, biology, and economics. As a free courseware, it is basically a collection of course materials placed on the Web. Lê (1999) produced a web-based courseware called 'Academic Language Courseware' for the teaching of academic genre. It consists of tutorials on key aspects of academic genre, communicative interaction with teachers and other learners, recommended readings, and on-line resources. This courseware is based on the concept and principles of a virtual class. MarketPlace (Resnick, 1996) is another type of courseware which has a problem-based orientation. It enables students to participate in economic simulations over the Internet, playing the roles of buyers and sellers in a virtual marketplace. This problem-based courseware includes online discussion facilities, designed to support not only economic deal-making among participants but also reflection and analysis of the economic patterns that arise from the interactions (Resnick, 1996).

Evaluation of educational courseware takes into account the following factors:

- *Flexibility*: A courseware should be flexible enough to cater for learners' learning styles and interests when they negotiate through their learning and to allow the learners to learn at their own pace.
- *Interactivity*: This includes textual and communicative interactivity. Textually, learners can explore various parts of the courseware, which are structure-based and content-based. Communicatively, a communication board is provided for discussion among participants.
- *Multimedia power*: A range of multimedia tools are embedded in the courseware for learners to construct their own materials and to access other sources.
- *Resources*: It is important for promoting independent learning. Reading materials include scanned articles from books and those available on the Web. Guided tours of various web sites dealing with specific issues or topics should be made available.
- *A wide range of learning experiences*: Apart from the content-based topics dealing with different aspects of a course, there should be different sub-components such as practical implications, problem-solving tasks, and self-testing.
- *Learners' evaluation*: The courseware should include a component to provide a user-friendly facility for students to contribute their evaluation and feedback.

In summary, taking into account the different ways in which software is evaluated by teachers on the basis of their perception of the roles of software in teaching and learning, the process of educational software evaluation is presented in the following diagram.



This diagram illustrates the main stages of software evaluation. According to Geissinger (1997), education software is produced for an educational purpose. This sounds very simple. However, evaluation of educational software is a complex task as it requires an understanding of the principles and perspectives of teaching and learning and how they are used to develop educational software.

A simple question for any educational software should be: 'Can this product actually teach what it is supposed to?' It is a simple question to ask, but often difficult to answer because the product may have so many beguiling features. It requires the evaluator to recognize his/her own view of the ways in which students learn, to relate that view to the learning objectives of that portion of the course and to determine how and whether those objectives are carried out in the software (Geissinger, 1997, p. 222).

CONCLUSION

Educational software has proliferated in virtually every academic discipline and at various levels from preschool to postgraduate studies. Educational software enables a

computer system to process data, including both the operating system and a variety of application programs for teaching and learning. With the rapid development of computer technology, software is increasingly powerful. It makes the task of software evaluation more challenging. Basically, two main educational perspectives underlie current software evaluation: instructionist and constructionist. The former is generally teacher-centred and the latter is learner-centred. It is worth noting that it is not strictly an 'either-or' commitment as these two theoretical views are not necessarily mutually exclusive in eduction. Software is powerful not because it is technologically superior but because it enables educators of different educational perspectives, to bring creative innovations into teaching and learning.

REFERENCES

Bain, J., & McNaught, C. (1996). Academics' educational conceptions and the design and impact of computer software in higher education. In C. McBeath & R. Atkinson (Eds.), *Proceedings of the Third International Interactive Multimedia Symposium: The learning super highway. New world? New worries?* (pp. 55-59). Perth, Western Australia: Promaco Conventions Pty Ltd.

CEISMC. (2006). Evaluation Tools. Retrieved 12 November, 2006, from http://mime1.marc.gatech.edu/MM_Tools/evaluation.html

Geissinger, H. (1997). Educational software: Criteria for evaluation. In R. O. R. P. R. Kevill (Ed.), *Proceedings of ASCILITE'97 on what works and why?* (Vol. 219-225). Perth, WA: Curtin University of Technology.

Hammond, N., Trapp, A., & McKendree, J. (1994). Evaluating educational software: a suitable case for analysis. Retrieved 15 September, 2006, from http://ctiwebct.york.ac.uk/LTSNCiPAbstracts/CiP94/Hammond.html

Henkel, S. (2006). Reflective metaphors. Retrieved 18 October, 2006, from http://www.bethel.edu/~shenkel/Curriculum/Metaphors/Metaphors.html

Kizlik, B. (2006, 10 August 2006). How to Write Learning Objectives that Meet Demanding Behavioral Criteria. Retrieved 19 November, 2006, from http://www.adprima.com/objectives.htm

Lakoff, G. (1995). Metaphor, Morality, and Politics or Why Conservatives Have Left Liberals In the Dust. Retrieved 10 November, 2006, from http://www.wwcd.org/issues /Lakoff.html

Lakoff, G., & Johnson, M. (1980). *Metaphors We Live By*. Chicago: University of Chicago Press.

Le, Q. (1999). *The Development and Evaluation of a Multimedia-based Instruction Package on Scientific Genre.* Unpublished PhD, Curtin University of Technology, Perth.

MIT. (2002-2006). MITOPENTOCOURSEWARE. Retrieved 19 October, 2006, from http://ocw.mit.edu/index.html

Papert, S. (1993). *The Children's Machine: Rethinking School in the Age of the Computer*. New York: Basic Books.

Resnick, M. (1996). Distributed Constructionism. *Proceedings of the International Conference on the Learning Sciences Association for the Advancement of Computing in Education* Retrieved 28 October, 2006, from http://llk.media.mit.edu/papers/Distrib-Construc.html

Squires, D., & McDougall, A. (1994). *Choosing and using educational software: A teachers' guide*. London: The Falmer Press.

Wilss, L. (1997). Evaluation of computer assisted learning across four faculties at Queensland University of Technology: Student learning process and outcomes. In R. O. R. P. R. Kevill (Ed.), *Proceedings of ASCILITE'97 on what works and why?* (pp. 636-642). Perth, WA: Curtin University of Technology.