Technology and Teacher Education Annual, 1997

Proceedings of *SITE 97* —
Eighth International Conference of the
Society for Information Technology and
Teacher Education (SITE)
Orlando, Florida; April 1-5, 1997

Volume I

AACE Association for the Advancement of Computing in Education

The Design, Evaluation and Usage of Educational Software

Netiva Caftori
Northeastern Illlinois University

Marcin Paprzycki
University of Texas of the Permian Basin

In recent years a proliferation of educational software can be observed in all school levels as well as at home and in the workplace. One of the important reasons for this situation seems to be that by interacting with a computer unsupervised learning is possible (more than three students per teacher with no educational guidelines; see Caftori, 1996). At home overburdened parents can leave their children unattended using a program that has a "educational" label and expect that learning may occur while playing. A similar situation may occur in schools, where supplementary learning can be furnished in a computer laboratory. Finally, in the professional arena many firms opt for training through computer based instruction as a cost-cutting measure. It is much cheaper and more flexible to provide employees with training software than to organize a workshop with a paid trainer.

Results of a recent study of educational software use in unsupervised settings were rather disturbing, showing that children did not learn what they were expected (Caftori, 1994a). It was also shown how this learning environment leads to the development of a gender gap (Caftori, 1994a, 1994b). Part of the problem can be traced to the insufficient emphasis on evaluation of the quality of educational software by the educators themselves (Paprzycki & Caftori, 1996). The aims of this paper are to summarize the possible effects of unsupervised usage of the educational software and to suggest a number of changes necessary to avoid future problems. These changes will be related to: a) educational software evaluation, b) technology-based lesson delivery and c) university curricula.

Educational Software

A typical example of software use in educational practice was presented by Caftori (1994a). She describes how, in a Junior High School, a number of computer terminals have been set up so that students can interact with the educational software and learn in an unsupervised mode. It should be stressed that the software installed is clearly designated as educational. It was found that this educational software does not play an educational role in many cases, or at least not the educational role it was intended to play.

An informative example is based on the *Oregon Trail* game. This game is a history simulation with the educational goal of introducing children to the life of covered-wagon travelers on their way from the Missouri River to Oregon in 1848. In its educational objectives it is suggested that this game induces students to make intelligent decisions based on a limited amount of data and consider alternative solutions when the circumstances suddenly change. It contains a number of problem-solving situations, like river crossing, money and food management and dealing with disease outbreaks. It is also supposed to teach students to arrange the data they have gathered into the "bigger picture" and to establish interrelations between the facts so that they can make appropriate decisions. The overall effect of all the decisions made determines the final outcome of the game. A number of pedagogical problems were observed (for a complete discussion see Caftori 1994a):

- Children concentrate on reaching the end of the trail as fast as possible without regard for their companions or oxen.
- Children take no time to visit the landmarks and learn their history.
- Children shoot animals for the sake of shooting becomes an objective in itself; the type of the terrain and the animals associated with it are not noticed and/or learned by the student.

A number of similar problems were observed when studying how students interacted with other games:

- using trial and error strategies instead of calculations in games like *Paper Plane Pilot* and *Wood Car Rally*,
- not playing *Where in the World is Carmen San Diego?* as the game takes a long time for completion,

- playing the game of *Odell Lake* with the goal of swallowing fish to enjoy the sound effects (instead of learning the predator-prey relationship and the food chain).

In addition to these general problems some interesting observations have been made related to the gender differences in approaching the games (for a detailed discussion see Caftori 1994a and 1994b). Some of these observations match those presented in Christie, (1996), Fryer (1995), Goldstein, Olivares, and Valmont (1996) and Underwood and Underwood (1996), where additional examples can be found.

- Girls were less visible in the computer laboratory and thus participated in the supplementary learning less often.
- Girls in early grades prefer word games (such as *Hangman*) to construction with geometric figures (such as *Mosaic*).
- Overall, boys like fast, shooting, fighting, or killing games involving battle or space ships while girls prefer slower games involving writing or school work.
- Even when playing the same game (*Oregon Trail*) girls have pursued different goals such as reaching the destination (the original goal of the game), or writing epitaphs on tombstones),
- Girls do not like software that does not allow them to quit a section in the middle; boys do not like software that does not provide them with an appropriate feedback.

Summarizing, even though students were interacting with the educational software, they were able to do it in such a way that at least some (if not all) of the specified educational objectives have been missed. Software attributes that were intended to attract children to the game (e.g. competitiveness, animations) actually diverted their attention from the objectives. The observational results confirm also that much software is designed to appeal to boys without consideration of the effect it has on girls (see also Huff & Cooper, 1987).

It is easy to specify what types of changes could have made the Oregon Trail game a more valuable educational tool. For example, when a poor decision is made by the student while traveling on the trail, the software should hint that a more efficient way may exist. When the student strays away from the set goal there should be a reminder provided about the major objective of the game. Similar suggestions can be easily made for the remaining games.

We were more interested in finding out why these deficiencies were not spotted when the games were evaluated, before the label "educational" or "good" was attached to them. We have searched for the criteria that are applied to evaluate educational software. To find the answer we have, initially, studied three issues of the *Technology and Teacher Education Annual* (Carey, Carey, Willis & Willis, 1993; Willis, Robin & Willis, 1994; Willis, Robin & Willis, 1995) *Proceedings of Society for Information Technology and Teacher Education*. Our assumption was that since this is one of the biggest conferences addressing the usage of technology in education, our findings will be representative of the state of the art in the area. Later, we have also looked into the most current Proceedings volume (Robin, Price, Willis & Willis, 1996). We were definitely surprised by our findings. First, there is almost no material related to the unsupervised learning. Second, we have located only a total of six papers related somewhat to the issue of educational software classification and evaluation: Byrum (1993), Byrum (1994), Maddux (1993), Paprzycki & Caftori (1996), Persichitte (1995) and Valmont (1994). This is especially astonishing while compared with the number of papers suggesting that teachers should write their own educational software. How are these teachers supposed to do a good job at it if they have no background in evaluating educational software and differentiating between good and bad to start with?

We should point out that in the past there have been numerous papers discussing software evaluation (see for instance Bitter & Wighton, 1987). That was before new powerful hardware and multimedia became widespread leading to the development of new generations of educational software. In Paprzycki & Caftori (1996), we have shown how Oregon Trail can pass the elaborate set of evaluation criteria proposed by Persichitte (1995) and none of the problems reported above can be predicted. However, the topic of evaluation criteria for the educational software does not seem to be very popular at one of the largest forums where such a discussion should take place. This fact, combined with the problems indicated with the educational software, lead us to believe that not only a discussion of these issues should be initiated, but also other areas require additional attention.

Educational Software Evaluation

Since one of the more important problems with the educational software seem to stem from the lack of modern criteria to judge its educational quality we would like to present two groups of issues that need to be taken into account when such a judgment is being passed. This list is very preliminary and should be treated as a starting point for future research.

Social Issues

As discussed above, there exist substantial differences between the ways that boys and girls interact with the educational software and inappropriate usage of such software can further deepen the gender gaps. This seems to suggest that the following issues should be considered:

a) Does the interaction involve hand-eye coordination, problem solving, verbal interaction, or interaction between the students themselves in a cooperative/ collaborative mode of learning?
b) Does the software provide a diversity across genders/ races among the leading characters?
c) Does the game perpetuate gender/racial stereotypes or prejudices?

Educational Objectives

This group of criteria is supposed to address the problem of children playing the game without reaching the expected educational outcomes.

a) Are the educational goals precisely defined by the software developers?
b) Does the software vendor provide any instruction about how the software can be used in-class and at-home so that the best results can be expected?
c) Does the software encourage reaching these goals by penalizing various misuses or providing guidance toward reaching the goals?
d) Does the software discourage trial-and-error type behavior while favoring decisions resulting from thought processes?
e) are the special or interactive effects overwhelming the objectives of the game?
f) Can the educational objectives or a reasonable subset be reached in a limited or prescribed time?

These two groups of criteria should be supplemented by detailed evaluations related to the particular medium and its usage in the classroom or unsupervised setting. For instance, there seems to be a body of experience mounting that may lead to the development of such evaluation criteria for CD-ROM's used in reading courses (see Goldstein et. al. 1996, Land & Taylor 1995, Matthew 1996, Underwood & Underwood 1996 and Valmont 1994).

Changes in the Classroom

The problems described above indicate that even if the quality of the educational software improves the teacher will still be needed in the classroom where computers will be used in a lab or studio environment. At the same time the teacher will have to work as hard, if not harder, to make sure the software employed is appropriate and is used in a manner that enhances the material to be learned. The basic work will be concentrated on developing lesson plans built around the software. Anyone who has observed a lesson delivered in the computer laboratory realizes immediately that such a lesson changes the position of the teacher. As soon as students start working on the computers their attention is diverted from the teacher and it is extremely difficult to get their attention back. As a consequence, the teacher is no longer at the front of the classroom as a center of attention and the class dissolves into units working either individually or in groups. Each such unit pursues knowledge independently and at a different pace. This means that the in-class situation resembles very closely the unsupervised learning. Therefore the lesson plans will have to address this new, different role of the teacher, as guide or helper instead of information provider. An extended discussion of this new role can be found in Caftori (1996).

Curricular Changes

Finally, changes aimed at addressing the problems indicated above should also appear in the university curricula. Our observations of the way that educational software is used singled out at least three areas where serious problems can surface. First, at home, where parents invest in the educational software and hope that this will help their children to succeed. Second, in schools, where undertrained teachers use educational programs in the unsupervised mode to furnish students with supplementary learning. Finally, among educational software developers, who may not be aware of what happens when their software is used in real-life situations by real-life learners. We would like to suggest that this indicates three corresponding areas where the introduction of computers into our everyday lives imposes the need for the curricular update: computer education for the general student population, teacher preparation (pre-service and in-service), and the computer science curriculum itself.

Computer Education for the General Student Body

It may be assumed that most of the students are current or prospective parents and thus they are (will be) the ones who (will) buy the educational software for their children. They should learn that the educational software is not a remedy for the lack of supervised learning and interaction with their children. They should be made aware of what the basic problems related to the unsupervised usage of educational software are. It seems that the most natural course for transferring this knowledge would be the Computer Literacy course. Unfortunately, currently this course is used primarily to teach basic application packages. The situation would change if the idea of "computers across curriculum" suggested in Paprzycki, (1996) and used in a few universities, were to be accepted. In this model the Computer Literacy course is being removed from the curriculum. The computer component is introduced to all courses offered in the curriculum. This solution can free a slot to offer a capstone course devoted to the social and ethical issues related to the usage of computers. In this course the considerations related to the usage of educational software could naturally find a home.

Computer Education in Teacher Preparation

Not all Colleges of Education provide the courses necessary for training teachers to integrate computers into

the curriculum (this should be distinguished from the courses devoted to the usage of technology in the classroom settings, which are relatively popular). The need for such an integration and preparation of "informationally literate" teachers has been very well summarized in Niederhauser (1996). Pressure must be exerted to include an Educational Software Methods course into the pre-service teacher preparation core curriculum for all teaching fields. This course should be also introduced and promoted as an important part of in-service teacher training. The content of this course should contain issues related to the in-class usage of educational software as well as unsupervised usage of educational software. A substantial component discussing criteria and methodology of educational software evaluation (including exercises) should also be included. (For a description of a course that could be modified to meet these objectives see Mitchell and Paprzycki, 1993, and the references cited there.)

It should be pointed out, that our proposal goes against suggestions similar to that of Loehr (1996) and Valde et. al. (1996). The first proposal is much too narrow and seems to suggest concentration on the mechanics of technology usage which is a remedial subject that should not be a part of the university curriculum. The second proposal is clearly misguided by suggesting that teachers should have knowledge of computing in a pseudo-programming language. This is not the type of knowledge that will be ever useful to the teachers as they need to know how to utilize the ready-to-use software only and will never be involved in any form of computer programming.

Computer Science Curriculum.

First, let us observe that typically most CS students are taught how to write software, how to write it fast, how to write it well (where well means without errors). Almost no attention is given to the content of the created product. In other words, CS students are as efficient in writing good quality educational software as in writing educational software that does not educate at all. They do as well in developing a good quality human-computer interfaces as creating a bad one. What is badly needed is a course that would concentrate on the variety of issues related to what can be called human-oriented software. The proposed topics could include a selection out of the following:

- computers as a medium of communication,
- elements of computer graphics,
- computers and society,
- psychology of human-computer interaction,
- design of computer-human interfaces,
- design of educational software,
- design of software for handicapped users,
- ethical issues of computer usage,
- gender issues in computer use.

Taking into account the important role computers play in our lives and how they have become an integral part of society, we would like to suggest that such a course should be required of every graduating Computer Science major interested in pursuing software design career. A special track may be offered or even a new minor or major concentration as is suggested by McGuire (1995). He describes how at the American University (Washington, DC) the Art and the Computer Science and Information Systems Departments have combined in a similar effort. It should be pointed out that, in most cases, this will be the role of the School of Education to pressure the Department of Computer Science to develop an appropriate course.

Conclusion

In this paper we have tried to suggest to the reader that what is labeled as "educational software" may not be very educational after all. We have also argued that issues related to the educational software evaluation seem not to be very high on the list of researchers in the pedagogical sciences. It is true that evaluating software is part of some existing teacher preparation courses; however, these courses themselves have not been recently evaluated and their outcomes assessed. These considerations led us to propose changes in three related areas: academic curricula, software evaluation and software in-class usage. Whereas our proposals are very preliminary, the point is to generate discussion and stimulate research in these areas.

Educational Software

Educational software products mentioned in this text are available from the Minnesota Educational Computing Consortium (MECC), an educational software developer and distributor from Minneapolis, MN.

References

Bitter, G.G. & Wighton, D. (1987). The most important criteria used by the educational software evaluation consortium. *The Computing Teacher*, 7-9.

Byrum, D.C. (1993). Selecting computer applications: A model for pre-service teachers. In D. Carey, R. Carey, D. Willis & J. Willis (Eds.), *Technology and Teacher Education Annual 1993* (pp.336-340). Charlottesville, VA: Association for the Advancement of Computers in Education.

Byrum, D.C. (1994). Educational software types: An interactive Hypercard tutorial. In J. Willis, B. Robin, & D. Willis (Eds.), *Technology and Teacher Education Annual 1994* (pp.454-458), Charlottesville, VA: Association for the Advancement of Computers in Education.

Caftori, N. (1994a). Educational effectiveness of computer software. *T.H.E. Journal*, 62-65.

Caftori, N. (1994b). Examination of computer software in relation to gender differentiation. *Journal of Women and Minorities in Science and Engineering, 1* (3), 237-252.

Caftori, N. (1996). *Give up Your Pedestal, but Don't Give up Your Lesson Plans*. [On-line]. Available http://www.ecnet.net/users/uncaftor/ped.txt

Carey, D., Carey, R., Willis, D. & Willis, J, (Eds.). (1993). *Technology and Teacher Education Annual 1993.* Charlottesville, VA: Association for the Advancement of Computers in Education..

Christie, A. A. (1996). Can technology help teachers and students down gender stereotypes? In B. Robin, J. Price, J. Willis, & D. Willis (Eds.), *Technology and Teacher Education Annual 1996* (pp.31-35). Charlottesville, VA: Association for the Advancement of Computers in Education.

Fryer, B. (1995, December). Sexism and kids' software, *PC World*, 406.

Goldstein, B., Olivares, E. & Valmont, W. (1996). CD-ROM storybooks: Children's' interactions. In B. Robin, J. Price, J. Willis, & D. Willis (Eds.), *Technology and Teacher Education Annual 1996* (pp. 110-111). Charlottesville, VA: Association for the Advancement of Computers in Education.

Huff, C. & Cooper, J. (1987). Sex bias in educational software: Effects of designers' stereotypes on the software they design. *Journal of Applied Social Psychology, 17* (3), 519-532.

Land, B.L. & Taylor, R. (1995). Planning and creating interactive, multimedia lessons for literature-based reading programs. In D. Willis, J. Robin, & J. Willis (Eds.), *Technology and Teacher Education Annual 1995* (pp.132-135) Charlottesville, VA: Association for the Advancement of Computers in Education.

Loehr, M. (1996). Top ten media competency recommendations by teachers for teachers training. In B. Robin, J. Price, J. Willis, & D. Willis (Eds.), *Technology and Teacher Education Annual 1996* (pp.474-476). Charlottesville, VA: Association for the Advancement of Computers in Education.

Maddux, C.D. (1993). Teaching teachers to evaluate software: A system and an example. In D. Carey, R. Carey, D. Willis & J. Willis (Eds.), *Technology and Teacher Education Annual 1993* (pp.212-215). Charlottesville, VA: Association for the Advancement of Computers in Education.

Matthew, K.I. (1996). The promise and potential of CD-ROM books. In B. Robin, J. Price, J. Willis, & D. Willis (Eds.), *Technology and Teacher Education Annual 1996* (pp.116-119). Charlottesville, VA: Association for the Advancement of Computers in Education.

McGuire, E. G. (1995). An undergraduate degree in multimedia design and development: Collaboration between departments of CS and IS and art. *Proceedings of ISECON '95*, 17-24.

Mitchell, T. & Paprzycki, M. (1993). Teaching new technologies: developing a computer literacy course for prospective teachers. *Journal of Computing in Small Colleges, 8* (3), 65-70.

Niederhauser, D.S. (1996). Information age literacy: Preparing educators for the 21st century. In B. Robin, J. Price, J. Willis, & D. Willis (Eds.), *Technology and Teacher Education Annual 1996* (pp.415-418). Charlottesville, VA: Association for the Advancement of Computers in Education.

Paprzycki, M. (1996) Computer literacy across curriculum. *Journal of Computing in Small Colleges, 11* (7), 94-103

Paprzycki, M. & Caftori, N. (1996). Educational software: Are we approaching it the right way? In B. Robin, J. Price, J. Willis, & D. Willis (Eds.), *Technology and Teacher Education Annual 1996* (pp.869-873). Charlottesville, VA: Association for the Advancement of Computers in Education.

Persichitte, K. (1995). Basic criteria for selecting and evaluating instructional software. In D. Willis, J. Robin, & J. Willis (Eds.), *Technology and Teacher Education Annual 1995* (pp.379-382) Charlottesville, VA: Association for the Advancement of Computers in Education.

Robin, B., Price, J., Willis, J., & Willis, D. (Eds.), *Technology and Teacher Education Annual 1996*. Charlottesville, VA: Association for the Advancement of Computers in Education.

Valde, R., Bower, R. & Thomas, R. (1996). Developing preservice teachers' computer competencies. In J. Price, J. Willis, & D. Willis (Eds.), *Technology and Teacher Education Annual 1996* (pp.604-607). Charlottesville, VA: Association for the Advancement of Computers in Education.

Valmont, W., (1994). Electronic children's books: the good, the bad, and the hype. In J. Willis, B. Robin, & D. Willis (Eds.), *Technology and Teacher Education Annual 1994* (pp.495-497), Charlottesville, VA: Association for the Advancement of Computers in Education.

Underwood, G. & Underwood, J. (1996). Gender differences in children's learning from interactive books. In J. Price, J. Willis, & D. Willis (Eds.), *Technology and Teacher Education Annual 1996* (pp.136-139). Charlottesville, VA: Association for the Advancement of Computers in Education.

Willis, J., Robin, B., & Willis, D. (Eds.), *Technology and Teacher Education Annual 1994*. Charlottesville, VA: Association for the Advancement of Computers in Education.

Willis, D., Price, J. & Willis, J. (Eds.) (1995). *Technology and Teacher Education Annual 1995*. Charlottesville, VA: Association for the Advancement of Computers in Education.

Netiva Caftori is in the Department of Computer Science, Northeastern Illinois University, Chicago, IL 60625 E-mail: n-caftori@neiu.edu

Marcin Paprzycki is in the Department of Mathematics and Computer Science, University of Texas of the Permian Basin, Odessa, TX 79762 E-mail: paprzycki_m@utpb.edu