An Evaluation of Competing Multimedia Software Packages & A Software Evaluation Model

Jing Yang & Rob Power

Education 531: Assessment of Software and Information Technology
Posted: February 20, 2003

Contents:

Introduction:

This paper provides a review of the authors' evaluations of two competing multimedia software packages, Tech4Learning's *MediaBlender 2.0* (Tech4Learning, 2002) and Knowledge Adventure's *HyperStudio 4.2.0* (Knowledge Adventure, 2000). The evaluations were carried out using a teacher evaluation checklist and student survey along with a software evaluation model developed by the authors (Yang and Power, 2003) as part of the course requirements for Education 531: Assessment of Software and Information Technology, at University College of Cape Breton. The two software packages were compared based upon their technical merits, as well as their abilities to help meet a predetermined set of high school English Language Arts curriculum objectives from the Atlantic Provinces Education Foundation English Language Arts Curriculum Guide, 10-12 (*APEF*, 2000). After reviewing the evaluations, this paper also takes a closer look at the effectiveness of the software evaluation model itself and the necessity of using such evaluation models when selecting software for educational purposes.

Section One: An Evaluation of Competing Multimedia Software Packages

According to the *ELA Software Evaluation Model*, the aim of the evaluation is to help educators evaluate the usefulness and reliability of software packages in helping to teach their students, meet or exceed curriculum standards, and improve the overall teaching and learning environment (Jing and Power, 2003). Though the model was developed using the senior high school English Language Arts (ELA) curriculum objectives outlined by the Atlantic Provinces Education Foundation (*APEF*, 2000), it can be used for a general purpose for other curriculum software evaluation. The authors used the printable software evaluation form (including software attributes checklist and curriculum objectives checklist) to evaluate *MediaBlender* (Tech4Learning, 2002) and *HyperStudio* (Knowledge Adventure, 2000).

In the software attributes evaluation section, there are six distinct categories covering everything from the appearance and user-friendliness of the software package to its actual educational content. The rating scale is from 1 - not present/unacceptable to 5 – exceptional.

MediaBlender and *HyperStudio* are two multimedia presentation tools. Their functions are similar as is the interface. According to the results of the evaluation form, the total score of *MediaBlender*'s attributes is 474, which belongs to the "excellent" range; while the total score of *HyperStudio*'s attributes is 471, which is the top score of the "good" range. The close score indicates the two software packages have similar attributes, but *MediaBlender* ranks a little higher. In the aesthetic value category, *MediaBlender* got a score of 44 and *HyperStudio* got a score of 46. *HyperStudio* thus has a more attractive appearance which means it is interesting or fun to use, while in the user friendliness category, *MediaBlender* got 70 and *HyperStudio* got 68.

For students, *MediaBlender* has a more user-friendly interface and the icons in it are easier to locate and use compared to *HyperStudio*. In the technical aspects category, *HyperStudio* got a higher score of 135 and *MediaBlender* got 129. Obviously, *HyperStudio*'s technical value is higher than *MediaBlender*. For example, *MediaBlender* has an ftp software link to website, but *HyperStudio* has more styles and actions to link to website. In *HyperStudio*, the user can use the function button of the mouse to add clip art when right clicking the mouse button, but in *MediaBlender* this choice is not available. *HyperStudio* also has the slide show function, which makes it more convenient and considerate. In the Extras manual, *HyperStudio* has many other functions. *MediaBlender* has fewer functions than *HyperStudio*, but it is easier for new users to use and it can export files to html format and the files are transmitted to the website via ftp directly. In *HyperStudio*, stack can be changed to webpage format, but it still needs .stk file and the target computer should have *MS ActiveX Gallery* installed. With *HyperStudio*, users can insert voice, sound and video easily into the stack to express their idea more clearly. But to use this software, some training work should be completed. It is not very easy for a rookie to use it. In general, *MediaBlender* is good for new users, while *HyperStudio* is better for advanced users.

In the adaptability/ flexibility and instructional value categories, the two software packages got very close scores, which indicates that both applications can be adapted for multiple uses, and the aspects of the both software packages can be integrated to meet curriculum requirements. As for the instructional value, both packages help to increase learning according to the score 46/44 in a score range from 12 to 60, but neither are working very effectively. In the final educational content category, *MediaBlender* got 129 and *HyperStudio* got 120. Here, *MediaBlender*'s ease of use helps to better meet specific curriculum objectives for students and increase specific outcomes.

Curriculum outcomes are what we are mainly looking for in both applications. In the evaluation form, we continue to use a rating scale from 1 to 5, rating each of the following specific ELA curriculum outcomes that are addressed by using the software. Again, though our model was developed using the senior high school ELA curriculum objectives outlined by the Atlantic Provinces Education Foundation, it can be used for a general purpose for other curriculum software evaluation. We have three specific categories related to ELA curriculum outcomes. These include speaking and listening skills, reading and viewing skills, and writing and other ways of representing skills. In each category we have corresponding criteria, these specific criteria will not be included in the final tally for the rating of the software, but they will help to give other teachers a better picture of the usefulness of the software package.

Both software packages have similar curriculum outcomes concerning students speaking and listening skills. *MediaBlender* and *HyperStudio* both can adequately examine others' ideas and synthesize what is helpful to clarify and extend their own understanding, as well as asking discriminating questions to acquire, interpret, analyze and evaluate ideas and information. *HyperStudio* works better on articulating, advocating and justifying positions on an issue or text in a convincing manner, showing an understanding of a range of opposing viewpoints. Both applications work well in interacting in leadership and support roles in a range of situations, such as complex purpose/procedure and different subject matter. They provide a good quality in adapting language and delivery for a variety of audiences and purposes in informal and formal contexts. But they are not flexible enough in responding to a wide range of complex questions and directions. *MediaBlender* is better than *HyperStudio* in addressing the demands of a variety of speaking situations and making critical language choices.

In the reading and viewing skills category, *MediaBlender* works better than *HyperStudio* according to the rating scores. *HyperStudio* is weaker when articulating students' understanding of ways in which information texts are constructed for particular purposes and making informed personal responses to increasingly challenging print and media texts and reflecting on their responses. They both help to select texts to support students' learning needs and range of special interests and they work exceptionally well in using cueing systems and a variety of strategies to construct meaning in reading and viewing complex and sophisticated print and media texts. *MediaBlender* and *HyperStudio* also support accessing, selecting and researching specific information to meet personal and learning needs and are good at showing the relationships among language, topic, purpose, context and audience. But both software packages have deficiencies in critically evaluating the information the student access and responding critically to complex and sophisticated texts.

In the writing and other ways of representing skills category, *HyperStudio* and *MediaBlender* got very close scores. They work exceptionally well in the following outcomes: make effective choices of language and techniques to enhance the impact of imaginative writing and other ways of representing; produce writing and other forms of representation characterized by increasing complexity of thought, structure and conventions; demonstrate understanding of the ways in which the construction of texts can create, enhance or control meaning; apply students knowledge of what strategies are effective for them as creators of various writing and media productions; use the conventions of written language accurately and consistently in final products; use technology to effectively serve their communication purposes; demonstrate a commitment to the skilful crafting of a range of writing and other representations and integrate information from many sources to construct and communicate meaning. Both *MediaBlender* and *HyperStudio* adequately evaluate the responses of others to their writing and media productions.

In general, the teacher's checklist indicates that *MediaBlender* and *HyperStudio* are both good software for presentation in teaching and learning. They each have good and insufficient aspects, which require more revision to make them more adaptable to classroom teachers and students.

The student survey form for the evaluation also reflects students' preference and priority on using both software packages. *HyperStudio* got a total score of 74, while *MediaBlender* got 83. The students showed more interest in *MediaBlender* and ranked it higher than *HyperStudio*. They rated *MediaBlender* better in criteria like ease of use, support/help and enjoyable than *HyperStudio*.

In summary, according to the software evaluation model, we can conclude that *MediaBlender* and *HyperStudio* are both good software packages for educational purpose. From the teacher's view, *HyperStudio* has more functions and technical supports so that it is better for advanced users. From the student's view, *MediaBlender* is easier and has a more user-friendly interface. It is simple and concise, so that it's better for new users and lower level students.

Section Two: An Evaluation of the Software Evaluation Model

The ELA Software Evaluation Model (Yang and Power, 2003), which was used to carry out the comparisons reviewed in the first section of this paper, was developed based upon a review of literature on educational theories, current and best practices in software evaluation, examples of other software evaluation models, and practical classroom teaching experience (Driscoll, 1994; Kemp, Morrison and Ross, 1998; Maddux, LaMont Johnson and Willis, 1997;

Niederhauer and Stoddart, 2001; Rieser and Kegelmann, 1994; Sharp, 2002; Shuell, 1986; and Stirling, n.d.). After having tested the model against competing software packages, a number of questions must now be answered. How effective was this particular model for evaluating software to meet high school English Language Arts curriculum objectives? Is it necessary to go to such lengths in order to select software for educational applications in the classroom, or at the district or provincial educational levels? Is it necessary to include students, and the student survey segment, in the software evaluation process? And what changes, if any, could be made to improve the overall effectiveness of the software evaluation model itself?

How effective was the *ELA Software Evaluation Model*?

The software evaluation model was highly effective in comparing the two multimedia software packages, and helping to determine which package would be the most appropriate selection for integration into the high school English classroom. Although it was impossible, given the time constraints of this project and the lack of resources such as a sample of high school English Language Arts students, to carry out the pretesting, immediate Post-testing, and Delayed Post-testing stages of the model, the teacher evaluation checklist and student survey stages proved very easy to apply to this situation. These stages of the model also provided a wide range of information on which to base an effective comparison (Reiser and Kegelman, 1994; Stirling, n.d.). The model provided a complete list of the Atlantic Provinces Education Foundation English Language Arts Curriculum Outcomes for grades 10 through 12 (*APEF*, 2000), making it very easy to select the range of outcomes that a teacher could expect to meet using any multimedia software package. This checklist also made it easy to cross-references the evaluators' ratings of the ability of each software application to meet predetermined curriculum outcomes, as well as to get a sense of the ability of the software to be applied in other contexts, to meet a broader range of outcomes (Niederhauer and Stoddart, 2001; Reiser and Kegelman, 1994; Sharp, 2002; Stirling, n.d.). Furthermore, the teacher evaluation stage of the model allowed for easy cross-referencing of a range of technical and aesthetic aspects of each software package, making it easy to determine which package had the most value in terms of technical capabilities, adaptability to different users and learning contexts, general appearance, and overall ease of use (*Ibid.*). While these qualities can easily be compared be reading technical reviews of the competing software packages, *the ELA Software Evaluation Model* allowed for a detailed overview of the effectiveness of the software from both an English teacher's and a student's perspective – information that could prove indispensable to any teacher's or administrator's software selection process (Ibid.).

Is it necessary to go to such lengths to select software for educational purposes?

Some form of meaningful information must be used as the basis for making decisions on the selection of educational software over competing software packages (*Ibid.*). While it is likely that some such decisions are made on the basis of personal preference, or inclusion of the application in packaged deals with software suppliers, such decision-making methods simply are not reliable when it comes to picking the right applications for educational contexts (*Ibid.*). Neither is reliance on an educator's intuition, scientific research studies, or the review of a software expert (Stirling, n.d.). Each of these methods has its drawbacks, which can only be overcome by carrying out a software evaluation process tailored to the needs of students and educators. Personal preference and the intuition of educators can be misleading, in that comparisons between competing software may not actually be made, and the educator's own level of technical competance may blind him or her to the usability of an application from the point-of-view of the ultimate end-user – the student (Niederhauer and Stoddart, 2001; Reiser and Kegelman, 1994; Sharp, 2002; Stirling, n.d.). Scientific research studies, while they appear to be the most reliable of information sources, are flawed in the rigidity of the scope of the actual study (Stirling, n.d.). The findings of the study may have very little transferability to different classroom learning contexts, or the needs of a varied student body. And the reviews of software experts may provide little information of value in educational contexts (*Ibid.*). Expert reviews can be flawed in their focus on the technical aspects of the software, the wide variety of checklist standards used by such reviewers, and the lack of understanding of educational contexts on the part of the reviewer. Teachers and educational administrators need information in order to make software selection decisions, and that information must not only be derived through reliable means, it must be applicable to their own educational contexts, and the needs of their schools and students (Niederhauer and Stoddart, 2001; Reiser and Kegelman, 1994; Sharp, 2002; Stirling, n.d.). The only way to get that information is to test the software against the educational objectives it will be used to meet and to try the software out on real students to see how receptive they are to the application, how easily they can use the application, and how well it actually helps them to learn (*Ibid.*). Failing to follow such a process does not always mean that teachers will wind up stuck with poor or useless software. But it does increase the likelihood that both funding and the time of teachers and students will be wasted, and that the most meaningful and effectively learning tools and experiences will be overlooked (*Ibid.*).

Is it necessary to include students in the software evaluation process?

Absolutely. Students are the ultimate end-users of any software packages chosen to be integrated into the curriculum, so those packages must be able to meet their specific needs and aptitudes (Stirling, n.d.). While teachers and administrators can effectively analyze some of the technical aspects of software, such as content accuracy, and system requirements, some input is needed from the students to really get a handle on the aesthetic qualities of the software, and exactly how easy the software is for students to use (*Ibid.*). Many teachers who will conduct software evaluations will have high degrees of technical competence, which may blind them to the ability of students to use a particular software package. For this reason, teachers may be too lenient when it comes to conducting evaluations. Students are often far more critical of both the user-friendliness and the aesthetic appeal of software than are teachers (*Ibid.*). And, usually, students have less inhibitions when it comes to speaking their mind about such things (*Ibid.*). However, there is another major reason why students must be included in the software evaluation process. Software evaluation models, such as the one developed by the authors, are specifically designed to help teachers and administrators determine whether software packages actually do what they are needed to do – that is, to help students learn (Niederhauer and Stoddart, 2001; Reiser and Kegelman, 1994; Sharp, 2002; Stirling, n.d.). The only way to effectively determine this is by finding out what students already know, and comparing that against what they know after using the software. It is impossible to compare students' knowledge before and after using the software without including students themselves in a test of the software (*Ibid.*).

What changes could be made to improve the effectiveness of the ELA Software Evaluation Model?

The *ELA Software Evaluation Model* is designed to be used with or without the Pretesting, Immediate Post-testing and Delayed Post-testing stages (Yang and Power, 2003).

However, the greatest wealth, and most reliable information is obtained when these stages are included. For this reason, removing these stages from the evaluation model would not be desirable. The checklists, themselves, also convey valuable information (Niederhauer and Stoddart, 2001; Reiser and Kegelman, 1994; Sharp, 2002; Stirling, n.d.; Yang and Power, 2003). Neither the teacher evaluation checklist nor the student survey are overly cumbersome to complete, and the model's four-part rating system provides several different basis on which to compare competing software packages (*Ibid.*). The only change that the authors would recommend to the model, at this point, would be to alter the final numerical rating scheme from the first section of the teacher evaluation checklist. The current rating scheme has a range from 118-590 points. Other software evaluation models have incorporated percentile rating schemes (ranging from 0-100 points), which the authors feel might be more appropriate for the purposes of such software evaluations (Doucette and Flynn, 2003; Geldras, 2003). The percentile ratings are generally more easily understood by anybody asked to conduct such an evaluation, or by anyone who picks up a completed evaluation form, and wants to know exactly how well a software package performed.

Conclusion

Software evaluation models are invaluable tools to teachers and administrators who must make decisions about what software packages to download or purchase for integration into the curriculum (Niederhauer and Stoddart, 2001; Reiser and Kegelman, 1994; Sharp, 2002; Stirling, n.d.). They provide a great wealth of relevant information, tailor-made for educational environments (*Ibid.*). In the case of the *ELA Software Evaluation Model*, that information is directed specifically towards evaluating software for integration into English Language Arts courses, allowing those who must decide which software to select to see exactly how well the software meets curriculum requirements, and to compare those results to the ratings of similar software (Yang and Power, 2003). In this paper, the authors have demonstrated the effectiveness of the *ELA Software Evaluation Model* in helping educators make an informed choice between two competing multimedia software packages. While both packages scored highly in many of the technical, aesthetic, and curricular areas, the software evaluation model helped to arrive at a conclusion that one of the packages, Tech4Learning's *MediaBlender* (Tech4Learning, 2002), was easier and more comfortable for students to use – making it an overall more appropriate selection for integration into a learning context.

References:

Doucette, Darlene and Tom Flynn (2003). Tom and Darleen's Software Evaluation Model. Education 531: Assessment of Software and Information Technology, Winter Semester, 2003, Assignment #1.

Driscoll, M.P. (1994). *Psychology of learning for Instruction*. Boston: Allyn and Bacon.

Foundation for the Atlantic Canada English Language Arts Curriculum (2000). Halifax, NS: Atlantic Provinces Education Foundation.

Geldres, Juan (2003). *Software Evaluation Model*. Education 531: Assessment of Software and Information Technology, Winter Semester, 2003, Assignment #1.

Kemp, J.E., Morrison, G.R. & Ross, S.M. (1998). *Designing Effective Instruction, Upper Saddle River NJ*, Prentice Hall.

Knowledge Adventure (2000). HyperStudio 4.2.0. Retrieved February 15, 2003, from ftp://ftp.hyperstudio.com/pub/preview/HS4iPreview.EXE

Maddux, C., LaMont Johnson, D, & Willis, J. (1997). *Educational Computing: Learning with Tomorrow's Technologies (2nd ed.).* Boston: Allyn & Bacon.

Niederhauer, Dale S., Stoddart, Trish (2001). Teachers' instructional perspectives and use of educational software. *Teaching and Teacher Education,* 17, 15-31.

Rieser, Robert A., and Harold W. Kegelmann (1994). "Evaluating Instructional Software: Review and Critique of Current Methods," Cyber Slang: The Ultimate Instant Online Encyclopedia. Retrieved January 20, 2003, from http://tecfa.unige.ch/staf/staf9698/mullerc/3/evalRev.html

Sharp, Vicki (2002). *Computer Education for Teachers: Integrating Technology into Classroom Teaching*, 4/e, University of California. McGraw. Hill, 2002. Retrieved Jan. 30, 2003 from http://www.mhhe.com/catalogs/solutions/gallery/bb/0072397721.html

Shuell, Thomas J. (1986). "Cognitive Conceptions of Learning", *Review of Educational Research* Winter 1986, Vol.56, No. 4, Pp.411-436.

Stirling, Deborah Lynn (n.d.). *Evaluating Instructional Software.* Retrieved January 17, 2003, from http://www.stirlinglaw.com/deborah/software.htm

Tech4Learning (2002). MediaBlender 2.0. Retrieved February 15, 2003, from http://www.tech4learning.com/mediablender/evaluation.html

Yang, Jing and Rob Power (2003). *The ELA Software Evaluation Model: A Software Evaluation Model for Atlantic Provinces Senior High School English Language Arts Teachers*. Posted February 1, 2003, to http://www.geocities.com/rlpmedit/winter03/ed531/assignment1.html