

Software Evaluation - A Pedagogic Solution

Sivakumar Alagumalai

Jonathan Anderson

The Flinders University of South Australia

V. Mala

Ministry of Education, Singapore

Numerous methods have been advanced in educational software evaluation.

tested at the School of Education. Details of setting-up and maintaining such a tool and implications for teachers / educators of IT, specifically with educational software are given. Teacher collaboration and the internet-based databases are also discussed.

Please send suggestions and queries to:

sivakumar.alagumalai@flinders.edu.au

Introduction

Instructional computing can be grouped into three categories: tool,

design, implementation where the resultant system is built, and

Although traditional checklist do look into content accuracy, pedagogical appropriateness, and user control, to name a few, the changing design of current software, especially multimedia titles, becomes a huge undertaking for the software selectors and evaluators. Software assessment which include 'selection, review and evaluation' can thus be a formidable task (McDougall and Squires, 1995). However in making the final evaluation and recommendation, specific outstanding sections in a "not so popular" software is rejected and eventually forgotten. Shuell and Schuecker (1989, p.147) thus indicate that "development and evaluation of educationally sound software is still in

line with Winship's (1988) statement th

innovative environment for teachers to explore lesson development. Complementing this online development and enhancing any teacher's lesson would be these softlets and its associated lesson plans and practical field-experience of the teacher. These valuable information provides not only the very network needed by teachers but also serves as a resource base for both practising and pre-service teachers. This paper advocates the development of such a resource for teachers – an infobase of softlets, and its associated practical processes and skills.

Lesson plans have been identified as important elements for successful teaching and learning processes (Saxe, 1992; Battle & Hawkins, 1996). These lesson plans developed and modified in real classroom situations are given further refinement when it is shared between the teaching community. Lesson plans may vary from culture to culture, but may have central connecting themes as trialled for the Science-On-line (SOL) project (Battle & Hawkins, 1996). In the SOL project, teachers were given control over the design and implementation of text content, image content and organisation and presentation of their lessons, but adhered to commonly agreed framework. The general plan adopted for the infobase incorporates structure of the SOL and the direction advocated by the

subheadings / fields for the infobase.

Level:

Subject:

Topic:

Software Title:

Publisher (URL):

Softlet Theme:

when multimedia moves from standalone workstation type to distributed multimedia or networked multimedia (Nicolo & Sapio, 1996).

Currently numerous multimedia packages developed by teachers through HyperStudio, Toolbook and Authorware are available on the internet through shockwave (Minoli & Keinath, 1994; Sapio & Nicolo, 1995). For these softlets available on the internet, the Uniform Resource Locators (URLs) would be indicated in the Publisher's field. The implications and potential for such an infobase are far reaching and enables the global expansion of teaching / learning, multimedia, technology and knowledge itself.

Infobase and supporting cognitive theories

Instructional software can be self running and could provide the necessary learning. Sweeters (1994) indicates that the currently available multimedia titles with full motion video, sound and extensive

have been developed by system scientists and hardcore software programmers, who may have limited knowledge of educational processes involved and its operation in the classroom. In light of this problem, Shuell & Scheckler (1989, p.136) argue that educational software development should be consistent with current knowledge about teaching and learning.

The infobase proposed here leans heavily on psychological perspective that combines cognitive theories of teaching and learning, with lesson plans being practical manifestations of these theories, and softlets which have been specifically identified, selected and trialled by practising teachers.

Sweeters (1994, p.47) purports that an efficient lesson plan, even though modelling the ASSURE model, should embrace the 'functions of learning' through the 'Events of Instruction' as advanced by Gagné and

Elicit expected performance

Provide feedback

Assess performance

Enhance retention

Figure 2. Robert Gagné's Events of Instruction

The introduction and use of softlets fit well in Events 1, 4 and 7.

Thus softlets can provide vital support at the pre-, instructional and post-instructional phases. The availability of synchronised lesson plan

As educational software evolves into a powerful force, the infobase provides the teacher these synergies within their grasps. The infobase brings together the advantage of an electronic learning / teaching systems (Sweeter, 1994), the power of electronically shared lesson plans (Battle & Hawkins, 1996) and at the same time coping with the demands of our learners (Shuell & Schueckler, 1989). The advancement of networking of professional teaching elements through the infobase further supports the notion of both evolving cyberspace pedagogies and methodologies advanced by Anderson and Alagumalai (1996).

Setting up the Teacher's Infobase

Common-Gateway-Interfaces (CGI) and Integrated Development Environment (IDE)-links are the backbone engines for the Teacher's Infobase. The HTTP protocol used by the internet is generally a one-way street, going from servers to clients. However, browsers can ask the server to display specific requests. Thus there is also the return requester path. CGIs function on this path (McComb, 1996). CGIs pass data in two ways: one is URL-

provided by the client, and there are no restriction to the number of character used." An example of these CGI functions is attached in Appendix A.

Tied to these CGI functions are the necessary IDE-links that passes commands from the CGIs to the databases that stores the information. The Microsoft's Access *.mdb database format was used as it provided several advantages over other database engines (Ga

CGI (Server)

S

E

R

V

E

R .EXE

C Database (*.mdb)

L

I

E

N

T HTML (Form/Panel)

Figure 4. Technical structure of Infobase

The Access database had fields corresponding to the subheadings of the infobase (Appendix B) and sat in a directory where the CGI functions were located. Website Professional Version (Beta) II was used to launch the infobase on the internet. The server software supports both the database engine and the compiled CGI functions.

Microsoft's Index Server (MIS) was used as the expert search engine.

The search mechanics of the MIS supports the basic query language. At its simplest, a query can be just a word or phrase of a field in the infobase. Figure 5 is a sample of the infobase search engine.

Figure 5. Infobase search engine

The MIS also allows for searches of combinations of various key fields using the operators AND, OR and NOT. This facility allows for refinement of searches and displays the required information on a HTML panel (Appendix C).

Implications of Teacher's Infobase and Conclusion

The infobase apart from providing the necessary information for teachers, facilitates use and re-use of software, which may otherwise have been rejected due to the wholistic software evaluation process and inadequate criteria. Furthermore, a dynamic, ever growing and continuously refined practical teacher information is developed to cope with information and technological explosion. Teachers need not be passive recipients of information but active contributors shaping methodology for the newer generation of learners.

In line with the argument advanced by Anderson and Alagumalai (1996), there is an inevitable shift in both pedagogies and methodologies. With institutions heading towards flexible delivery, it is timely that useful teacher's supporting information be made available. The infobase is a humble direction proposed and trials of full implementation have been successful. The success lies not in conquering technological capabilities, but in winning over the interests of teachers.

Apart from providing the best form of resource for teachers, the infobase would also be an important asset for researchers, especially for those doing comparative work in curriculum and information relating to teaching and learning. Cross cultural and between country comparisons would be facilitated by this dynamic infobase. In line with softlets being the main focus of this database, it would mean shifting the paradigm from teacher as software-user to that of software-designer. The infobase would thus put teachers in command of the direction of future educational software development. Indeed the infobase is a force to be reckoned with.

References

Alessi, S.M., & Trollip, S.R. (1991). *Computer-based Instruction: Methods and Development*. Englewood Cliff, NJ: Prentice Hall.

Anderson, J., & Alagumalai, S. (1996). From text-based pedagogy to cyber-based methodology. Paper presented at the 10th Joint Conference of the Australian Association for Research in Education and the Singapore Educational Research Association, Singapore (25-29 November 1996).

Battle, R., & Hawkins, I. (1996). The study of emerging teacher practices in internet-based lesson plan development. *Journal of Science Education and Technology*, 5(4), pp.321-342.

Blease, D. (1986). *Evaluating Educational Software*. London: Croom Helm.

Diptino, V., & Turner, S. (1995). Zapping the hypermedia zoo – Assessing students' hypermedia projects. *The Computing Teacher*, 22(7), pp. 8-11.

Eklund, J. (1995). Cognitive models for structuring hypermedia and implications for learning from the world-wide-web. [Online].

Available: [_ HYPERLINK](#)

<http://aysweb95/papers/hypertext/eklund/index.html>

__<http://aysweb95/papers/hypertext/eklund/index.html>_, (19/3/96)

Gagné, R.M., & Briggs,

Learning, 9, pp. 222-228.

McComb, G. (1996). JavaScript Sourcebook. New York: John Wiley & Sons, Inc.

McDougall, A., & Squires, D. (1995). A critical examination of the checklist approach in software selection. Journal of Educational Computing Research, 12(3), pp. 263-274.

McNeil, S. (1996). A practitioner validated list of competencies needed for courseware authoring. Technology and Teacher Educational Annual, Association for the Advancement of Computing in Education, pp. 564-570.

