Integrated Interactive Science Software: A New Role for Teachers

Authors:

Dr. Gregory MacKinnon, Ph.D., Assistant Professor, School of Education, Acadia University, Wolfville, NS, B0P 1X0, (phone: 902–585–1385), (fax: 902–585–1071)
gregory.mackinnon@acadiau.ca, http://ace.acadiau.ca/fps/educ/home.htm

Kevin Deveau, B.Sc., B.Ed.
Therese Forsythe, B.Sc., M.Ed. (Classroom teacher)

Abstract

This paper offers a model for teachers to create their own software based on their regional curricular outcomes. The implementation of this model has profound implications for the classroom structure and promotes a truly facilitative role for the teacher.

Computer Technology in the Science Classroom

Recently I polled 500 teachers of science (grades 7–12) to ask them how they used computer software in their classrooms. The most popular innovation (79%) was the advent of the computer probe. These so-called CBL (computer-based laboratories) and MBL (microcomputer-based laboratories) are quickly gaining in popularity because they allow science teachers to promote higher-order thinking (Kurubacak, 1998) in ways that teachers could never access before. Students can gather data very quickly using temperature, motion, pH and pressure probes and thus spend the majority of their time critically considering their findings. Teachers have developed innovative activities that utilize ‘retro-analysis’. In these exercises, students are supplied with graphs and asked to generate them using experiments with the probes. This represents a very different way of thinking for children and early indications (Berger et al, 1994) are that many children are learning more effectively in these settings.

This article is less about recognised technological successes in science classrooms, and more about the potential of computers for promoting a facilitator role for teachers in the coming millennia. In this same survey teachers complained repeatedly that they spend a lot of money on pieces of software that have singular utility in certain aspects in their curriculum. Applications that are great for that one class of the year, but software that really misses the mark in ever other respect. This ‘poor value’ for the limited technology dollar, arose because most software just didn’t fit the teacher’s curriculum outcomes. How can teachers respond to this?

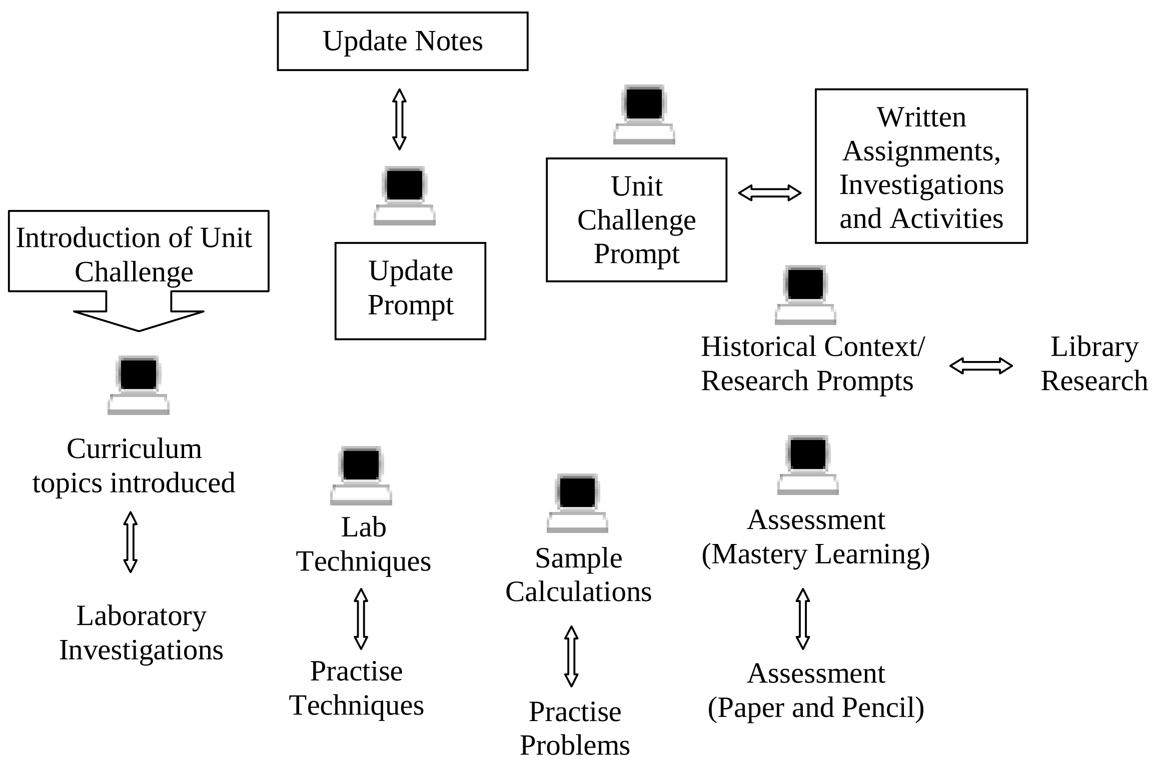
Opportunities

More than ever before, teachers have access to simple software tools that allow them to design software for their classrooms either on their own or in school software teams. Advances in user-friendly multimedia and a corresponding drop in the cost of these tools has made such approaches accessible for all teachers. Perhaps the most critical progress has been made in the area of creating ‘non-linear environments’. In these software settings, students are able to use such things as buttons and hypertext to access a wide variety of media. Simple software development is no longer the exclusive realm of so-called 'early-adopters' of computers. Thanks to products like Powerpoint and Hyperstudio teachers can begin to formulate truly 'integrated' software. These multimedia tools are relatively inexpensive and have become increasingly more powerful but not at the cost of making them 'user-unfriendly'. They still maintain very simple user interfaces with comfortable learning curves. Those teachers who want more flexibility may consider more powerful tools such as Authorware or Toolbook. Though these are more expensive products they remain fairly simple to use at the entry level.

A Learning Model

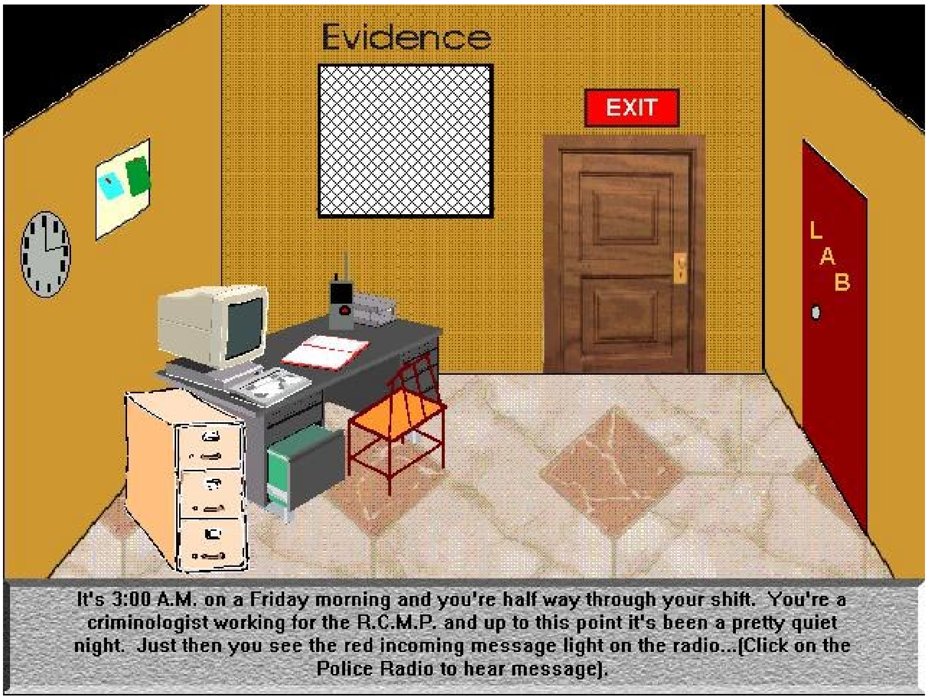
What does it mean to create interactive and integrated software? Interactive simply means students are entering data into the computer as it prompts them for information. This information can be anything from numerical data for calculations to multiple choice questions to test content knowledge. In our research, 'integrated' means developing software that directs students through the content in their specific curriculum. Our learning model for such integration is shown in Figure 1. At the onset of the unit, students are given a problem that draws on the process skills and content knowledge which will be developed throughout the unit. This we call the unit challenge and it is usually given to students in paper copy at the beginning of the study. Students then progress through the learning cycle periodically reflecting back (at the prompt of the computer) to the unit challenge as they construct new meanings throughout.

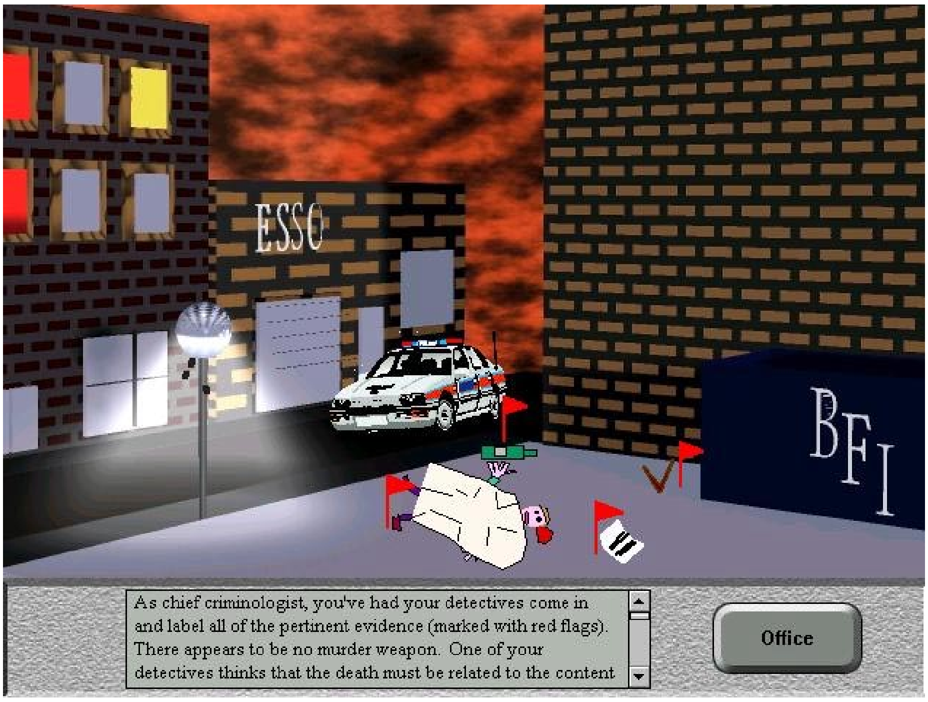
Figure 1: A Model for Integrating Computers into Curriculum



The computer serves a number of functions in this model none of which represents a stand-alone approach. This important aspect of the model addresses the lack of computer resources in many classrooms (Dockterman, 1997). Students can enter the unit learning cycle at a variety of nodes and thus two or three computers in a classroom can be quite sufficient to direct the learning of several groups of students. The computer then acts as a director of classroom learning activities. As can been seen in Figure 2-4 below, the software introduces the unit with a simple ‘omni–directional’ hypertext menu or graphic interface with selectable hot spots.

Figure 2: Software Developed for a Science Plus Unit on Solutions





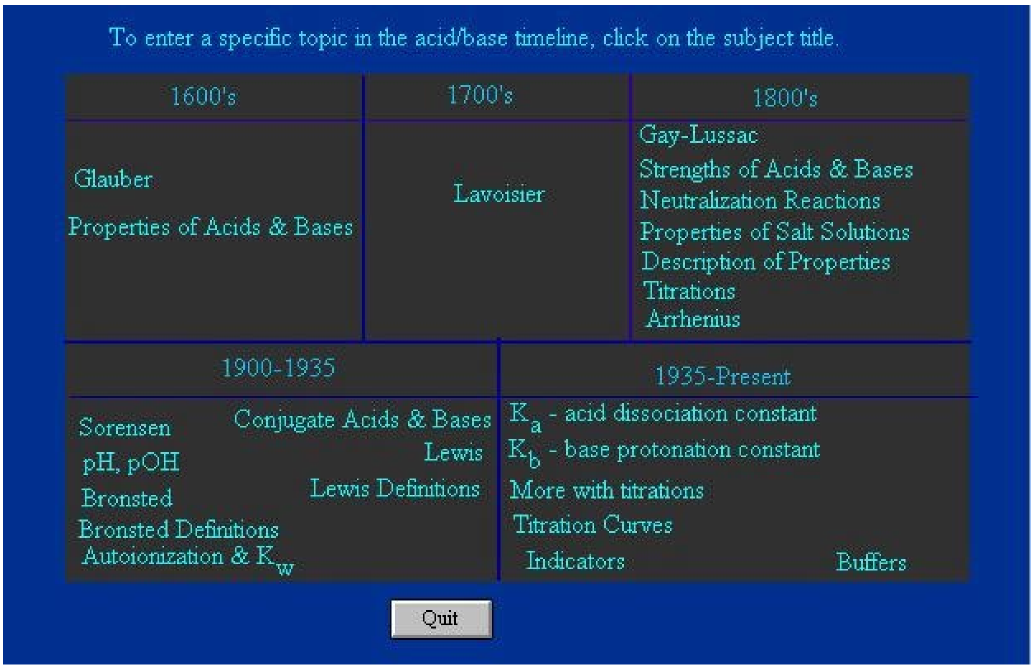
Samples of Interactive, Intergrative Teacher–Generated Software

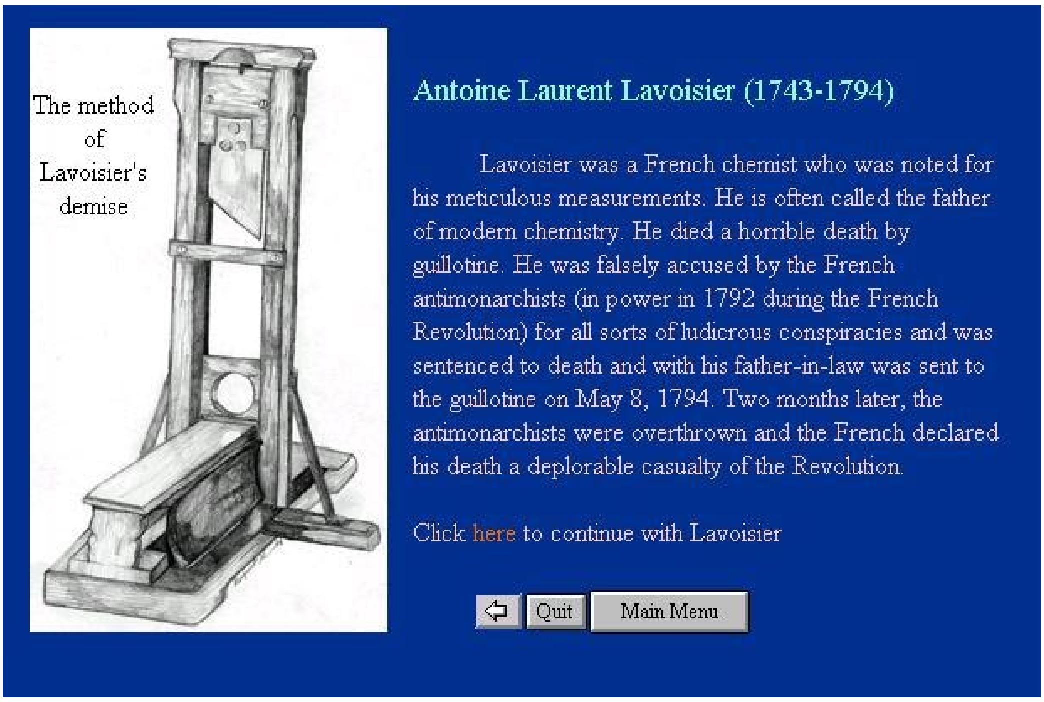
In the solutions software (Figure 2) students are presented with an interface that gives them access to a variety of activities through two interfaces. Their overall challenge is to determine who committed a crime. The computer serves as a database while students, through a series of solution-related activities (chromatography, suspensions etc.) narrow the suspect list to determine the culprit. In the office they can visit the crime scene for evidence, go to the lab to examine the evidence, enter data into the computer and open the filing cabinet to examine personal files on the criminals. At the crime scene each piece of evidence is flagged and here students begin their research by clicking the data they wish to follow up on. Students in a single classroom can be working simultaneously on different parts of the unit through this interface.

The software shown in Figure 3 was created for a six-week Grade 12 unit on acid-base chemistry. The overall unit challenge for the students was to solve a complex authentic problem of acidification of lakes in their region. As research groups they were to submit a proposal for environmental cleanup for the system. As they mastered various aspects of the unit content they would gradually build up the resources to respond to this challenge. This type of situated learning (Carr et al, 1998) has the potential to teach students to be better solvers of ill-structured problems (Spiro et al, 1992).

The timeline interface in this software helps to contextualise (McFadden, 1991) the development of acid-base theory. Though the menuing system allows access to all aspects of the unit at any time, students have preferred (MacKinnon & Forsythe, 1999) to address content in a linear sequence. In preparing software, teachers will find that some topics by nature are additive and thus student entry at different nodes of the learning cycle (Figure 1) is less practical. Because developing historical context of science helps students to situate their learning (MacKinnon, 1996) we have purposefully included (Figure 3) historical vignettes in this software.

Figure 3:Software Developed for a Grade 12 Chemistry Unit on Acids and Bases

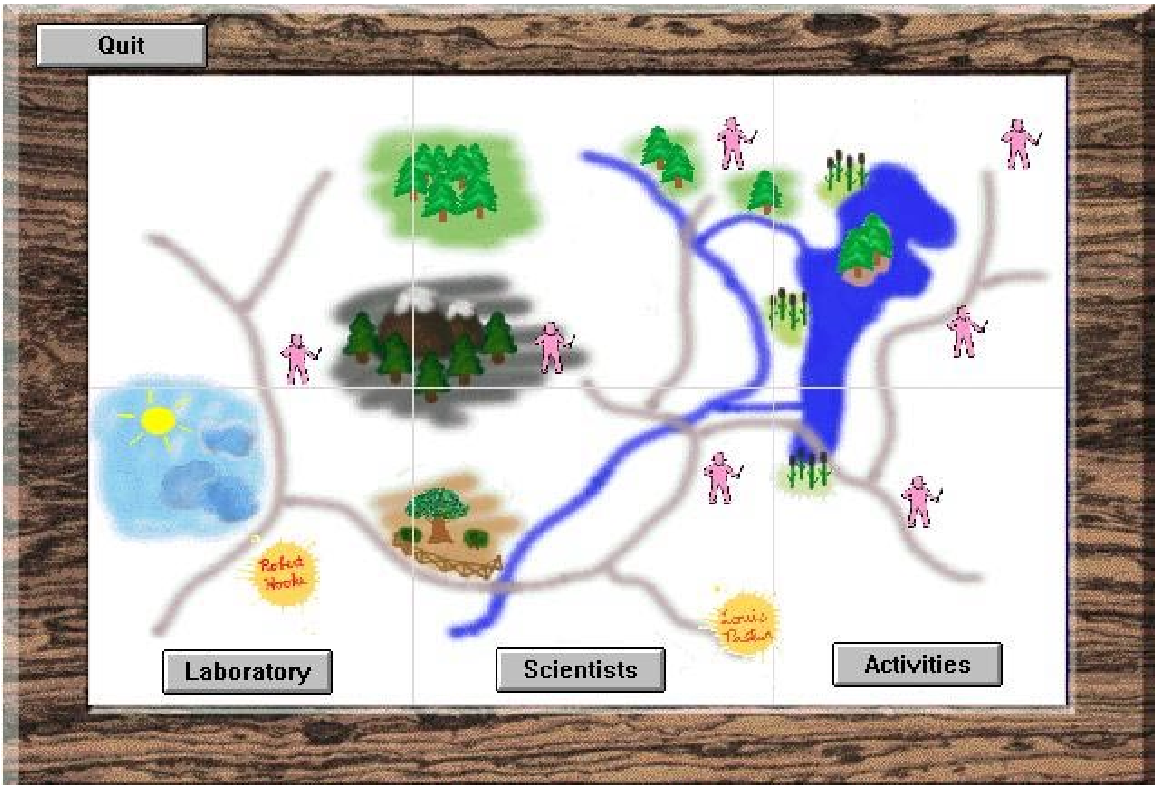


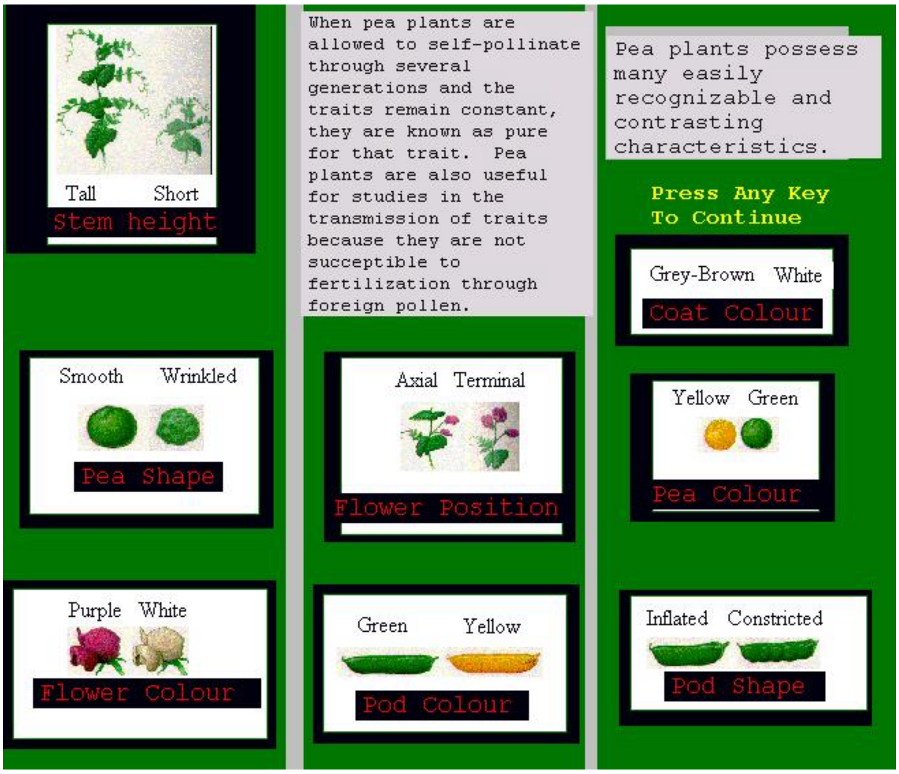


The software example shown in Figure 4 is unique in several ways. The content being developed is the science of the cell and the program could be used at a variety of grade levels. This large project has multiple components and is part of an initiative (Fasano & Brown, 1992) to adapt secondary school curriculum materials for use in inclusive classrooms. From the map interface (Figure 4) students take a historical journey that follows the science of the cell. Each quadrant accesses different scientists (Hooke, Mendel, Fleming, etc.) and their corresponding experiments. The student entering the program is mentored through a series of investigations at and away from the computer. Non-productive experiments and program paths have been included with corresponding positive reinforcement in an effort to emulate real science. In our program, the experiments of Mendel (Figure 4) are a particularly good example of how multimedia can be used to simulate cross-pollination experiments. Throughout these exercises students develop higher-order thinking skills as they analyse data and formulate concepts en-route to meaningful understandings.

This project has been particularly insightful for the authors in terms of teacher led software development. The quadrant approach was promoted because of the inherent flexibility it allowed to add compartmentalised content regarding the cell at a later date. We found that it was far more productive to have a teacher team work on a single component of this project rather than assign individuals to each of the quadrants (i.e. one person working on Hooke, one person on Mendel etc.). Secondly it should be noted that creating software for students of special needs is a unique challenge. In considering each and every student-computer interaction we diversified the interface. To mention a few, this was accomplished by use of larger buttons, text and screens, scrolling text combined with recorded and voice-read text, simplified data input schemes, readily repeated audio and video clips and printable screens.

Figure 4: Software Developed for the Cell and the Experiments of Mendel





Implications for Teachers

In our software designs, the computer may; introduce historical content, pose numerical problems, animate or simulate complicated processes, show video clips of demonstrations, send students away to perform laboratory investigations or library research or perhaps do classroom activities and assignments. Integrating computers into the curriculum in this way ensures a very active classroom setting. The organisational framework helps students to regulate their own learning (Shin, 1998) and work at their own pace. In our classroom observations, we have found that this maintains high motivation and on-task behaviours. In a recent pilot study of this model (MacKinnon & Forsythe, 1999), students were seen to utilise the teacher in very different ways. The co-operative group would field most trivial questions amongst themselves. Students used the computer screen as a teaching aid to explain concepts to the their peers. Generally, questions directed to the teacher were for clarification of instructions or more frequently, to re-explain a difficult concept. Overall students felt that the random access of the teacher for help was improved greatly as a result of this model. These systems have many benefits that parallel ‘centres approaches’ made popular in the elementary classroom. Intrinsic to this setting is considerable preparation as the teacher offers up all aspects of the unit on the first day! Subsequently the teacher is not preoccupied with the delivery of knowledge but moreover the engagement of that knowledge with students.

There has been an abundance of research (Berger et al, 1994) on the impacts of computers in science classrooms. Pedagogically-sound models that move beyond the techno-romantic era in considering the realities of limited resources in real classrooms, are most likely to emerge as useful technologies. Through teacher teamwork, classroom settings can be created that make efficient use of the available technology while ensuring specific curriculum outcomes are met.

References

Berger, C.F., Lu, C.R., Belzer, S.J. & Voss, B.E. (1994). *Research on the Uses of Technology in Science Education.* In D. Gabel (Ed.), Handbook of Research on Science Teaching and Learning, (pp. 466–490). New York, NY: MacMillan Publishing.

Carr, A.A., Jonassen, D.H., Litzinger, M.E. & Marra, R. M. (1998). *Good Ideas to Foment Educational Revolution: The Role of Systemic Change in Advancing Situated Learning, Constructivism and Feminist Pedagogy*. Educational Technology, 5–15.

Dockterman, D.A. (1997). *Great Teaching in the One Computer Classroom.* Watertown, MA: Tom Snyder Productions.

Fasano, J. & Brown, M. (1992). *Facilitating Inclusive Secondary Classrooms Through Curriculum Adaptation*. Exceptionality Education Canada, 2(1), 155–179.

Kurubacak, G. (1998). *Critical Thinking and Educational Computer Applications*. Annual Proceedings of the Society for Information Technology. (From Teacher Education, 1264–1267)

McFadden, C. (1991). *Towards an STS School Curriculum*. Science Education, 75(4), 457–469.

MacKinnon, G. & Forsythe, T. (1999). *Pilot Study in Progress*.

MacKinnon, G. (1996). *Biographical Science*. The Science Teacher, 63(2), 43–45.

Shin, M. (1998). *Promoting Student's Self-regulation Ability: Guidelines for Instructional Design*. Educational Technology, 38(1), 38–44.

Spiro, R.J., Feltovich, P.J., Jacobsen, M.J., & Coulson, R.L. (1992). *Cognitive flexibility, constructivism & hypertext: Random access instruction for advanced knowledge acquisition in ill-structured domains*. In T. Duffy & D. Jonassen (Eds.), Constructivism and the Technology of Instruction. Hillsdale, NJ: Erlbaum.